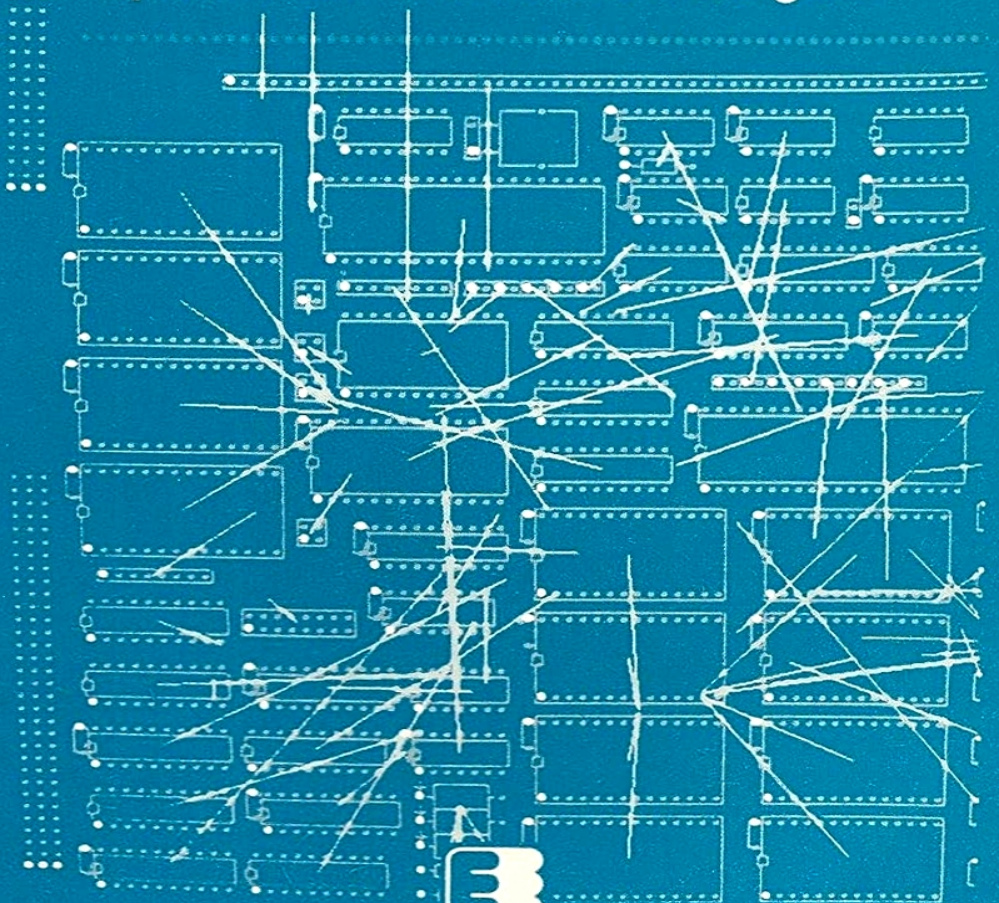
enet enst
COLLECTION TECHNIQUE ET SCIENTIFIQUE DES TÉLÉCOMMUNICATIONS

PRATIQUE DES CIRCUITS LOGIQUES

par J.-M. Bernard et J. Hugon



EYROLLES

DANS LA MEME COLLECTION ET CHEZ LE MEME EDITEUR

Des mêmes auteurs : — *De la logique câblée aux microprocesseurs.*
3 - Méthodes de conception de systèmes.
4 - Application des méthodes de synthèse.

J.M. BERNARD — *Conception structurée des systèmes logiques.*

GRINSEC — *La commutation électronique.*
1 - Structure des systèmes spatiaux et temporels
2 - Logiciel. Mise en œuvre des systèmes.

M. FELDMANN — *Théorie des réseaux et systèmes linéaires.*

H. SAUVAGEOT — *Radarmétéorologie.*
Télédétection active de l'atmosphère.

E. GELENBE, G. PUJOLLE
— *Introduction aux réseaux de files d'attente.*

J. DUPRAZ — *Probabilités, signaux, bruits.*

L. MICLET — *Méthodes structurelles pour la reconnaissance des formes.*

C. VASSALO — *Théorie des guides d'ondes électromagnétiques. (en 2 tomes).*

Sous la coordination de :

R. SOARES, J. GRAFFEUIL, J. OBREGON
— *Applications des transistors à effet de champ en arseniure de gallium.*

COLLECTION TECHNIQUE ET SCIENTIFIQUE DES TÉLÉCOMMUNICATIONS

PRATIQUE DES CIRCUITS LOGIQUES

par

Jean-Michel BERNARD **Jean HUGON**

Ingénieurs en chef
au Centre National d'Études des Télécommunications

TROISIEME EDITION
Revue et corrigée

EYROLLES

61, boulevard Saint-Germain – 75005 PARIS
1987

Si vous désirez être tenu au courant de nos publications, il vous suffit d'adresser votre carte de visite au :

Service « Presse », Editions EYROLLES,
61, Boulevard Saint-Germain
75240 PARIS CEDEX 05,

en précisant les domaines qui vous intéressent. Vous recevrez régulièrement un avis de parution des nouveautés en vente chez votre libraire habituel.

« La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40). »

« Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal. ».

© Editions EYROLLES et CNET-ENST, 1987

AVANT-PROPOS

Les connaissances nécessaires au concepteur de systèmes logiques comportent plusieurs volets.

Bien sûr, il faut un peu de théorie (algèbre de Boole): la pratique montre cependant qu'en ce domaine, un bagage restreint suffit. Pour concevoir des systèmes, nul besoin de mathématiques sophistiquées mais plutôt d'idées et d'expérience. Nous avons d'ailleurs constaté que plus le niveau d'abstraction des outils théoriques s'accroît, plus la taille des systèmes concevables avec ces outils diminue.

Bien sûr, il faut aussi connaître les fonctions logiques des opérateurs.

Mais surtout il faut être conscient que les composants ne se comportent pas uniquement comme des opérateurs booléens. Ils possèdent aussi des aspects électriques et surtout temporels que l'on ne doit pas négliger sous peine de courir à la catastrophe.

Cet ouvrage s'attache ainsi à présenter les composants logiques actuels sous tous les aspects utiles au concepteur de système (théorique, logique, électrique et temporel), sans toutefois descendre à un niveau descriptif trop fin, qui n'intéresserait plus le concepteur de système mais celui des composants.

Les chapitres 1, 2 et 3 traitent de questions théoriques: algèbre binaire, codage de l'information et traitements arithmétiques.

Les chapitres 4, 5 et 6 étudient les aspects électriques et tempo-

rels des circuits. On y analyse tous les paramètres usuels des circuits logiques et leur utilité pratique pour le concepteur de système. Quelques exemples concrets servent d'applications.

Les chapitres 7 et 8 passent en revue les fonctions combinatoires courantes (portes, égalité, comparaison, décodage, multiplexage, etc...) et montrent comment les associer pour étendre leur taille tandis que le chapitre 9 les utilise dans des applications de synthèse.

Les chapitres 10, 11, 12 et 13 sont consacrés aux fonctions séquentielles simples d'abord (bistable, bascule, monostable, oscillateur), puis aux fonctions séquentielles complexes (registre de mémorisation, comptage, décalage).

Dans le chapitre 14 on étudie différents montages séquentiels classiques et on calcule leurs performances à partir de leurs paramètres temporels.

Les chapitres 15, 16, 17 couvrent le domaine des mémoires à semi-conducteur, en dégageant d'abord leurs propriétés générales, en indiquant ensuite leurs applications et en réalisant une carte mémoire à partir des composants étudiés dans l'ouvrage.

Enfin le chapitre 18 donne quelques méthodes pour réaliser des systèmes séquentiels simples, les automates étant ensuite implémentés avec des circuits intégrés.

Nous pensons que ce livre constituera une introduction permettant au lecteur d'étudier la synthèse de systèmes réels et complexes comme les ingénieurs en rencontrent dans leur vie professionnelle. A cet effet, nous leur conseillons la lecture de "De la logique câblée aux microprocesseurs" tomes 3 et 4. par J.M. BERNARD et J. HUGON aux Éditions Eyrolles.

Les auteurs

TABLE DES MATIÈRES

AVANT-PROPOS V

CHAPITRE PREMIER — **Éléments d'algèbre binaire** 1

1. *Variables et fonctions binaires* 1
2. *Les trois fonctions de base en algèbre de Boole* 2
   - 2.1. La fonction NON 2
   - 2.2. La fonction OU (inclusif) 2
   - 2.3. La fonction ET 3
3. *Extension des fonctions ET et OU* 4
4. *Autres fonctions courantes de deux variables* 4
   - 4.1. Fonction NI, ou NON-OU, ou NOR (Not OR) 4
   - 4.2. Fonction NON-ET ou NAND (Not AND) 5
   - 4.3. Fonction OU exclusif 5
   - 4.4. Fonction identité 6
   - 4.5. Fonction logique quelconque complètement définie 6
   - 4.6. Fonction logique quelconque incomplètement définie 7
5. *Propriétés des opérateurs ET, OU, NON* 8
6. *Écritures canoniques d'une fonction logique* 9
   - 6.1. Fonctions produits 10
   - 6.2. Fonctions quelconques écrites en somme de produits 10
   - 6.3. Fonctions sommes 11
   - 6.4. Fonctions quelconques écrites en produit de sommes 12
7. *Simplification de l'écriture des fonctions logiques* 13
8. *Simplification algébrique* 13
9. *Simplification par la méthode de Karnaugh* 14
   - 9.1. Construction des tableaux 14
   - 9.2. Passage du tableau de vérité au tableau de Karnaugh 17
   - 9.3. Méthode de Karnaugh 18
   - 9.4. Application à une fonction de quatre variables 18
   - 9.5. Application à une fonction de cinq variables 20

10. *Méthodes algorithmiques* 21
11. *Symboles graphiques associés aux fonctions booléennes* 22
12. *Logigramme d'une fonction booléenne* 24

CHAPITRE DEUX — **Codage et protection de l'information** 27

1. *Divers aspects d'une même information* 27
2. *Représentation des nombres entiers* 27
3. *Comptage dans les différents systèmes de représentation* 28
4. *Changements de base de numérotation* 30
   4.1. Soustractions successives 30
   4.2. Divisions successives 31
5. *Représentation des caractères* 32
   5.1. Code ASCII 32
   5.2. Autres codes 33
6. *Changement de code* 35
   6.1. Principe 35
   6.2. Transcodage binaire vers décimal codé binaire 35
7. *Les codes détecteurs d'erreurs* 36
   7.1. Contrôle de parité ou d'imparité 37
   7.2. Codes *p* parmi *n* 39
8. *Les codes correcteurs d'erreurs* 39
   8.1. Code de Hamming 40
   8.2. Parité croisée 44

CHAPITRE TROIS — **Codage des nombres et opérations arithmétiques** 45

1. *Principe des opérations arithmétiques* 45
2. *Les différents modes de représentation des nombres* 47
   2.1. Nombres non signés 47
   2.2. Nombres signés sous la forme $\pm$ \|N\| (sign-magnitude) 47
   2.3. Nombres signés en complément à 2 48
   2.4. Nombres fractionnaires en virgule fixe 50
   2.5. Nombres en virgule flottante 51
3. *Addition et soustraction de deux nombres binaires non signés* 52
   3.1. Principe de l'addition 52
   3.2. Principe de la soustraction 53
   3.3. Étude des conditions aux limites 53
   3.4. Traitement du dépassement 55
   3.5. Soustraction ramenée à une addition 56
4. *Addition et soustraction de deux nombres binaires entiers en complément à 2* 57
   4.1. Addition et soustraction 57
   4.2. Étude des conditions aux limites. Débordement 58
   4.3. Traitement du débordement 60
5. *Addition, soustraction de nombres en virgule fixe* 62
6. *Addition, soustraction en virgule flottante* 62
7. *Algorithmes de la multiplication* 63
   7.1. Multiplication et division par des puissances de 2 de nombres non signés 63
   7.2. Multiplication et division par des puissances de 2 de nombres signés en complément à 2 63

7.3. Algorithme de la multiplication de deux nombres non signés ... 64
7.4. Algorithme de multiplication directe de nombres signés en complément à 2 ... 66
7.5. Amélioration des performances de la multiplication. Algorithme de Booth ... 70
7.6. Multiplication ramenée à des nombres ± |N| ... 73

CHAPITRE QUATRE — **Réalisation électrique des fonctions logiques** ... 77

1. *Présentation des fonctions logiques en boîtier* ... 77
2. *Les familles de circuits intégrés* ... 80
   2.1. Définition d'une famille logique ... 81
   2.2. Les familles logiques couramment utilisées ... 81
3. *Les familles TTL (Transistor-Transistor Logic)* ... 83
4. *Familles logiques utilisées* ... 84
5. *Fonctionnement d'une porte TTL standard* ... 84
   5.1. Schéma ... 84
   5.2. Fonctionnement ... 85
   5.3. Table de vérité. Convention logique ... 86
6. *Porte à sortie collecteur ouvert* ... 87
   6.1. Commande d'une charge externe ... 88
   6.2. Fonction ET câblé ... 88
   6.3. Transmission d'une information parmi n ... 90
7. *Porte à sortie 3 états "tri-state"* ... 91
8. *Porte à entrées trigger de Schmitt* ... 93

CHAPITRE CINQ — **Les paramètres électriques et temporels des circuits logiques** ... 97

1. *Introduction* ... 97
2. *La tension d'alimentation et la température de fonctionnement* ... 98
3. *Niveaux de sortie et d'entrée d'une porte* ... 98
4. *Courants de sortie ou d'entrée d'une porte standard* (sortie totem-pole) ... 100
5. *Courant de sortie pour une porte à collecteur ouvert* (open collector) et trois états (tri-state) ... 101
   5.1. Sortie à collecteur ouvert ... 101
   5.2. Sortie trois états ... 102
6. *Paramètres d'entrées d'une porte "trigger de Schmitt"* ... 103
7. *Autres paramètres électriques* ... 104
   7.1. Tension et courant d'écrêtage ... 104
   7.2. Courant d'entrée à la tension d'entrée maximum ... 105
   7.3. Courant de court-circuit ... 106
   7.4. Courant d'alimentation et puissance consommée ... 107
8. *Paramètres temporels des circuits combinatoires et séquentiels* ... 108
9. *Caractérisation d'une impulsion* ... 108
10. *Temps de propagation* ... 109
11. *Temps d'ouverture et de fermeture d'une porte "tri-state"* ... 110
12. *Paramètres temporels spécifiques aux circuits séquentiels* ... 110
13. *Utilisation des paramètres électriques et temporels* ... 111
    13.1. Paramètres $V_{OH}, V_{OL}, I_{OH}, I_{OL}, V_{IH}, V_{IL}, I_{IH}, I_{IL}$ ... 112
    13.2. Paramètres $I_{OZH}, I_{OZL}, I_{OL}, V_{IR}, I_I, I_{OS}, I_{CC}$ ... 114

13.3. Paramètres temporels de temps de propagation : $t_{PHL}, t_{PLH}, t_{PHZ}, t_{PLZ}, t_{PZH}, t_{PZL}$ 115
13.4. Paramètres temporels F, t_W, t_{su}, t_h, t_r 118
13.5. Variation des paramètres 119
14. *Exemples de paramètres électriques et dynamiques* 120

CHAPITRE SIX – **Exemples d'utilisation des paramètres électriques** 125

1. *Connexion de portes en "ET" câblé à collecteur ouvert* 125
1.1. Relation entre m, p, R_L pour le niveau haut 125
1.2. Relation entre m, p, R_L pour le niveau bas 126
1.3. Commande de plusieurs charges 127
1.4. Commande d'une seule charge 129
2. *Connexion de portes "tri-state" en commun* 131
2.1. Calcul au niveau haut 131
2.2. Calcul au niveau bas 132
2.3. Commande de plusieurs charges 132
2.4. Commande d'une seule charge 133
3. *Polarisation des entrées* 134

CHAPITRE SEPT – **Fonctions combinatoires usuelles et circuits associés** 137

1. *Introduction* 137
2. *Les opérateurs élémentaires* 138
2.1. Les types d'opérateurs 138
2.2. Caractéristiques électriques des opérateurs 138
3. *Comparaison entre informations* 139
3.1. Comparaison à zéro ou décodage à zéro 139
3.2. Égalité entre deux informations 140
3.3. Comparaison arithmétique de deux nombres A et B 141
4. *Transformation de codes* 144
4.1. Notions 144
4.2. Fonction de décodage binaire 144
4.3. Décodage binaire – Sept segments 148
4.4. Fonction d'encodage binaire prioritaire 149
4.5. Transcodage binaire décimal 150
5. *L'aiguillage d'information* 151
5.1. Généralités 151
5.2. Circuit multiplexeur 152
5.3. Circuit démultiplexeur 155
6. *Calcul et contrôle de parité : générateur de parité* 156
6.1. Principe 156
6.2. Le générateur de parité 74 LS 280 156
7. *Échange d'information par des bus* 157
8. *Circuits arithmétiques* 160
8.1. Équations logiques d'un additionneur 160
8.2. Réalisation d'un additionneur à retenue série 161
8.3. Additionneur à retenue anticipée : 74 LS 83 A 162
8.4. Bloc de calcul arithmétique et logique 166
8.5. Multiplieur 167
9. *Paramètres des circuits combinatoires complexes* 168

Annexes du chapitre sept 171

CHAPITRE HUIT – **Extension des fonctions combinatoires** 183
1. *Les extensions des fonctions combinatoires* 183
2. *Les extensions des fonctions ET, OU* 183
3. *Extension de la fonction de décodage* 185
4. *Extension de la fonction multiplexage* 187
5. *Extension de la fonction encodage* 188
6. *Extension de la fonction de comparaison arithmétique* 190
7. *Extension de la fonction égalité* 191
8. *Mise en cascade de bloc de calculs* 193
   8.1. Propagation série des retenues 193
   8.2. Propagation entièrement parallèle des retenues 194

CHAPITRE NEUF – **Conception des systèmes combinatoires** 199

1. *Méthode de conception* 199
2. *Étude d'un système comportant des fonctions connues* 202
3. *Étude d'un système simple* 203
   3.1. Spécification d'un distributeur de boisson 203
   3.2. Analyse 203
   3.3. Table de vérité 204
   3.4. Choix des composants 206
4. *Étude d'un système complexe* 220
   4.1. Spécification 220
   4.2. Analyse 221
   4.3. Codage des groupes d'eb 222
   4.4. Remarque sur les solutions retenues 225
5. *Comparaison entre les différentes méthodes de réalisation* 225

CHAPITRE DIX – **Les fonctions séquentielles simples** 229

1. *Comportement des systèmes combinatoires et séquentiels* 229
2. *Les fonctions séquentielles simples* 230
3. *Le bistable RS* 231
   3.1. Schéma 231
   3.2. Fonctionnement statique 232
   3.3. Étude du schéma du bistable RS réalisé avec des portes NOR 232
   3.4. Fonctionnement dynamique du bistable réalisé avec des portes NOR 234
   3.5. Remarques sur le fonctionnement dynamique 235
   3.6. Bistable RS réalisé avec des portes NAND 236
4. *Le bistable D* 237
5. *Utilisation des bistables* 238
6. *Rétroactions dans un système séquentiel* 239
   6.1. Limite d'emploi des bistables 239
   6.2. Les rétroactifs nécessitent des circuits complexes 241
7. *La bascule JK* 243
   7.1. Schémas symboliques 243
   7.2. Fonctionnement statique 244
   7.3. Diagramme des temps de fonctionnement 245
   7.4. Table de vérité d'une bascule JK et table de transition 245

7.5. Équation d'une bascule JK 247
7.6. Exemple de bascule JK 247
8. *La bascule D* 251
8.1. Schéma symbolique 251
8.2. Fonctionnement statique 252
8.3. Diagramme de fonctionnement 252
8.4. Équation et table de transition de la bascule D 253
8.5. Exemple de bascule D 253
9. *Le monostable* 254
9.1. Schéma symbolique 254
9.2. Fonctionnement 254
9.3. Diagramme des temps de fonctionnement 255
9.4. Domaine d'emploi 256
10. *L'astable ou oscillateur* 256
Annexes du chapitre dix 259

CHAPITRE ONZE — **La fonction de mémorisation** 263

1. *Les fonctions séquentielles complexes usuelles* 263
2. *Les commandes communes d'un registre* 266
2.1. Principe 266
2.2. Charge présentée par une commande 266
2.3. Priorité entre commandes 267
2.4. Commande avec trigger de Schmitt 267
3. *Les modes d'affichage* 267
3.1. Registres à commande d'affichage par état 267
3.2. Registres à affichage sur front d'horloge 269
4. *Les modes de sorties* 273
4.1. Sorties des bascules ou bistables 273
4.2. Sorties avec amplificateur séparateur 274
4.3. Sélection des sorties et interfaces trois états 274
5. *Exemple de registre intégré : le 74 LS 374* 275
6. *Applications des registres de mémorisation* 277
6.1. Systèmes sans automodification 277
6.2. Systèmes avec automodification des contenus registres 280
Annexes du chapitre onze 283

CHAPITRE DOUZE — **La fonction de comptage** 289

1. *Structure des compteurs binaires* 289
2. *Synthèse des compteurs binaires sans calcul* 290
2.1. Synthèse d'un compteur binaire synchrone 290
2.2. Synthèse d'un compteur binaire asynchrone 292
3. *Synthèse d'un compteur binaire par le calcul* 294
4. *Le comptage binaire incomplet* 295
5. *Le décomptage binaire* 297
6. *Diagramme des temps d'un compteur* 298
6.1. Diagramme général 298
6.2. Les états transitoires 298
7. *Mise en cascade des compteurs* 300
7.1. Compteur asynchrone 300
7.2. Compteur synchrone 301

8. *Le compteur programmable* 303
8.1. Principe 303
8.2. Compteur asynchrone avec remise à zéro asynchrone 303
8.3. Compteur synchrone avec commandes synchrones 304
8.4. Compteur synchrone avec commandes synchrones et asynchrones 307
9. *Propriétés générales des compteurs intégrés* 309
9.1. Comptage synchrone et asynchrone. Décomptage 309
9.2. Structure et taille 309
9.3. Les commandes d'un compteur 309
9.4. Les sorties 310
10. *Exemples de compteurs intégrés* 310
10.1. Le compteur binaire asynchrone 74 LS 93 310
10.2. Le compteur binaire asynchrone 74 LS 163 A 311
10.3. Le compteur-décompteur 74 LS 169 A 315
11. *Les applications du comptage* 316
11.1. Comptage d'événements 316
11.2. Division de fréquence 317
11.3. Exploitation des états d'un compteur. Parasites de décodage ... 317
11.4. Réalisation de cycles d'horloge 320
11.5. Réalisation de temporisation 322
11.6. Adressage d'éléments binaires 322
11.7. Registre de mémorisation 323
11.8. Rétroactions dans un compteur. Cycle incomplet 323
Annexe du chapitre douze 326

CHAPITRE TREIZE — **La fonction de décalage** 333

1. *Définition de la fonction de décalage* 333
2. *Les différents types de registres à décalage* 334
2.1. Registres à entrée série et sortie série (fig. 13.2.a.) 334
2.2. Entrée série, sortie parallèle (fig. 13.2.b.) 334
2.3. Entrée parallèle, sortie série (fig. 13.2.c.) 335
2.4. Entrée parallèle, sortie parallèle (fig. 13.2.d.) 335
2.5. Registre bidirectionnel (fig. 13.2.e.) 336
3. *Les différents types de décalages utilisés* 336
3.1. Décalage logique à droite ou à gauche 336
3.2. Décalage circulaire (rotation) à droite ou à gauche 336
3.3. Décalage numérique (ou arithmétique) 337
4. *Prélèvement d'information dans un registre à décalage* 337
5. *Principales caractéristiques des registres intégrés* 338
5.1. La fonction horloge 338
5.2. Les commandes 338
5.3. Les entrées-sorties 339
5.4. Organisation interne d'un registre à décalage 340
6. *Exemples de registres intégrés* 341
6.1. Registre à 4 eb 74 LS 194 341
6.2. Registre à 8 eb 74 LS 299 à entrées-sorties communes 344
6.3. Le décalage combinatoire 347
7. *Application du décalage* 349
7.1. Division de fréquence 349

7.2. Traitement série de l'information 350
Annexe du chapitre treize 351

CHAPITRE QUATORZE — **Assemblages séquentiels calculs temporels** 357

1. *Introduction* 357
2. *La bistable anti-rebond* 357
3. *Affichage en 1 ou 0 d'un bistable ou d'une bascule* 359
   3.1. Bistable RS 360
   3.2. Bascule JK 360
   3.3. Bascule D 361
4. *Transfert d'une information dans un point mémoire* 363
5. *Bistables à entrées multiples* 364
6. *Paramètres temporels des bascules* 365
   6.1. Rappel des paramètres temporels 365
   6.2. Calcul de la fréquence maximale de fonctionnement 366
   6.3. Temps de préaffichage et de maintien 367
   6.4. Durée des signaux 369
7. *Fréquence maximale de fonctionnement d'un compteur* 370
8. *Prélèvement d'information dans un registre* 371
9. *Réalisation de cycles d'horloge* 372
10. *Réalisation d'une temporisation* 373
11. *Réalisation d'un monostable logique programmable* 374

CHAPITRE QUINZE — **Les mémoires. Propriétés générales** 377

1. *Capacité et accessibilité* 377
2. *Les grandes catégories de mémoires* 378
   2.1. Les mémoires à semi-conducteur 378
   2.2. Les mémoires magnétiques 378
   2.3. Les mémoires à bulles magnétiques 379
3. *Technologies des mémoires à semi-conducteur* 379
   3.1. Technologie MOS ou Metal Oxide Semi conductor 379
   3.2. Technologie CMOS ou MOS Complémentaire 380
   3.3. Technologie bipolaire 380
4. *Classement des mémoires à semi-conducteur selon l'utilisation* 380
   4.1. Généralités 380
   4.2. La mémoire vive ou RAM (Random Access Memory) 381
   4.3. La mémoire morte ou ROM (Read Only Memory) 381
   4.4. La mémoire programmable par l'utilisateur ou PROM (Programmable Read Only Memory) 381
   4.5. La mémoire reprogrammable (REPROM ou EPROM) 382
   4.6. Les mémoires ROM effaçables électriquement (EEPROM) 382
   4.7. Mémoire FIFO ou file 382
   4.8. Mémoire LIFO ou pile 383
5. *Le point mémoire* 383
   5.1. Cellule statique d'une mémoire vive 383
   5.2. Cellule dynamique d'une mémoire vive 383
   5.3. Cellule de mémoire ROM 384
   5.4. Cellule de mémoire PROM 384
   5.5. Cellule de mémoire EPROM 384
   5.6. Cellule de mémoire FIFO 384

6. *Organisation des cellules mémoires* 384
6.1. Structure matricielle 384
6.2. Structure linéaire 385
7. *Accès aléatoire et séquentiel* 386
8. *Capacité et format d'une mémoire* 387
9. *Vitesse d'une mémoire* 388
10. *Consommation d'une mémoire* 388
10.1. État actif 389
10.2. État passif 389
10.3. État de veille 389
11. *Alimentation* 390
12. *Le brochage* 390
12.1. Normalisation du brochage 390
12.2. Réduction du nombre de broches 391
13. *Les paramètres électriques et temporels* 392
14. *Compatibilité entre mémoires* 393
15. *Intégrité de l'information* 393
16. *Description du fonctionnement des mémoires* 394
17. *Quelques qualificatifs courants concernant les mémoires* 395
18. *Réalisation de mémoires de grande capacité* 395
18.1. Assemblages de boîtiers à structure matricielle 395
18.2. Mémoire FIFO de grande capacité 396
18.3. Précaution de réalisation des mémoires de grande capacité 397
19. *Généralités sur les cycles de fonctionnement des mémoires* 397
19.1. État inactif, état de veille 397
19.2. Cycle de lecture 398
19.3. Cycle d'écriture 399
19.4. Cycle de lecture – Modification – Écriture 400
19.5. Cycles de lecture, écriture en mode page (fig. 15.13) 401
19.6. Cycle de rafraîchissement 402
19.7. Programmation 403
19.8. Cycle de programmation 403
19.9. Effacement 403
19.10 Récapitulation des cycles 403
20. *Les réseaux logiques programmables* 404
20.1. Réseaux combinatoires 404
20.2. Réseaux séquentiels programmables 409
Annexe du chapitre quinze 411

CHAPITRE SEIZE – **Application des mémoires** 417

1. *Stockage des programmes et des données associées* 417
2. *Tables de données fixes* 417
3. *Remplacement de registres dans une machine* 418
4. *Mémoire d'interface entre sous-ensembles* 418
5. *Transcodage* 420
6. *Machine séquentielle simple* 421
7. *Mémoire PROM utilisée en générateur de fonctions* 422
8. *Réalisation des mémoires dans un système* 422
8.1. Mémoires intégrées dans une logique de gestion 422
8.2. Mémoires organisées en cartes indépendantes 423

CHAPITRE DIX-SEPT – **Assemblage des boîtiers mémoire** 429

1. *Introduction* 429
2. *Brochage du boîtier* 429
3. *Principe d'exploitation* 430
4. *Cycle de lecture* 431
5. *Cycle d'écriture avancé* (EARLY WRITE CYLE) 432
6. *Cycle de lecture-modification-écriture* 434
7. *Tableau des paramètres* 435
8. *Spécification de la carte mémoire* 437
9. *Constitution de la matrice mémoire* 438
10. *Résolution des conflits de bus* 442

CHAPITRE DIX-HUIT – **Conception des systèmes séquentiels simples** 445

1. *Méthode de conception* 445
2. *Système simple composé de fonctions connues* 447
3. *Système simple composé de fonctions quelconques : détection synchrone de fronts* 448
   - 3.1. Synthèse avec tableau de vérité 448
   - 3.2. Diagramme des temps du montage 449
4. *Système simple composé de fonctions quelconques : générateur de caractères* 451
   - 4.1. Spécification 451
   - 4.2. Principe de réalisation 451
   - 4.3. Calcul d'une séquence sans répétition de caractère 452
   - 4.4. Cycle avec répétition 454

BIBLIOGRAPHIE 455

Chapitre premier

ÉLÉMENTS D'ALGÈBRE BINAIRE

1. VARIABLES ET FONCTIONS BINAIRES

Les machines qui traitent l'information sont constituées de circuits qui ne possèdent que deux états électriques stables. A partir des variables d'entrée binaires, les machines calculent des fonctions de sortie binaires.

Ces variables ou fonctions binaires se rencontrent également dans la vie courante. Par exemple, un automobiliste va prendre la décision de ralentir :

- Si un panneau ralentir est présent ET si l'automobiliste est discipliné.
- OU si un voyant de défaut grave s'allume sur son tableau de bord.

Si ces conditions ne sont pas réalisées, l'automobiliste ne ralentit pas. Ralentir, dans ces conditions est bien une fonction binaire dépendant de variables binaires (le panneau est présent ou non, l'automobiliste discipliné ou non, etc...).

Pour étudier d'une manière systématique ces fonctions de variables binaires, il existe une algèbre binaire, très différente de l'algèbre classique, qui porte le nom d'algèbre de Boole, ou algèbre booléenne, imaginée par George Boole, mathématicien anglais (1815-1864). Nous nous proposons d'examiner dans la suite de ce chapitre les notions d'algèbre de Boole nécessaires et suffisantes pour l'étude et la conception de circuits logiques, sans prétendre à la rigueur mathématique, que le lecteur pourra trouver dans des ouvrages spécialisés.

2. LES TROIS FONCTIONS DE BASE EN ALGÈBRE DE BOOLE

2.1. La fonction NON

Elle porte aussi les noms de négation, inversion, ou de complément. Elle se note:

$f(x) = \overline{x}$ (prononcer x barre).

Si x vaut 0, NON x vaut 1 et si $x = 1$, NON x vaut 0.

D'une manière générale, on définit une fonction booléenne par un *tableau de vérité*, qui donne la valeur de la fonction (0 ou 1) pour l'ensemble des combinaisons des variables. Dans le cas présent, la fonction inversion de x est définie par le tableau de la figure 1.1.

	variable	Fonction NON	
	x	$\overline{x}$	
Valeurs possibles de x	0	1	Valeurs correspondantes de Non x
	1	0	

Fig. 1.1. — *Tableau de vérité de la fonction inversion.*

2.2. La fonction OU (inclusif)

Cette fonction de deux variables $f(x,y)$, qui porte également les noms de réunion, union, addition logique ou somme logique est notée couramment avec le signe + de l'arithmétique traditionnelle.

$f(x,y) = x + y$ (prononcer x ou y).

Certains, pour éviter la confusion avec l'arithmétique écrivent:

$f(x,y) = x \vee y$

Généralement, cette distinction est superflue et l'on sait parfaitement, en fonction du contexte, s'il s'agit d'une véritable addition arithmétique ou de la fonction OU. Dans la suite, nous adopterons la première notation. Le tableau de vérité de la figure 1.2 donne les valeurs de $f(x,y) = x + y$ pour les

quatre valeurs de x et y. Cette fonction vaut 1 si l'une ou l'autre des variables x, y ou les deux valent 1.

	Variables		Fonction OU x+y	
	x	y		
Valeurs des variables	0	0	0	Valeurs de la Fonction
	0	1	1	
	1	0	1	
	1	1	1	

Fig. 1.2. — *Tableau de vérité de la fonction OU.*

2.3. La fonction ET

Il s'agit également d'une fonction $f(x,y)$ à deux variables, appelée aussi produit logique ou intersection. On la note couramment comme le produit algébrique avec un point entre les variables.

$f(x,y) = x.y$ (prononcer xy)

ou même sans point s'il est évident que x et y sont deux variables séparées.

$f(x,y) = xy$

Certains auteurs, pour distinguer le ET du produit algébrique notent :

$f(x,y) = x \wedge y$

Nous utiliserons dans la suite les deux premières notations. La figure 1.3 donne le tableau de vérité de la fonction ET. La fonction vaut 1 si x et y valent 1.

Variables		Fonction ET xy
x	y	
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 1.3. — *Tableau de vérité de la fonction ET.*

3. EXTENSION DES FONCTIONS ET ET OU

Les fonctions ET et OU s'étendent facilement à plus de deux variables. Ainsi, la fonction ET de trois variables x, y, z vaut 1 si x et y et z sont simultanément en 1. De même, la fonction OU de trois variables vaut 1 si $x = 1$ quelque soit l'état de y et z, ou $y = 1$ quelque soit l'état de x et z, ou $z = 1$ quelque soit l'état de y et x.

Le tableau de vérité de la figure 1.4 décrit ces fonctions étendues. Le principe est le même pour plus de trois variables.

variables			Fonction ET	Fonction OU
x	y	z	xyz	x+y+z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig. 1.4. — *Tableau de vérité des fonctions ET, OU à trois variables.*

4. AUTRES FONCTIONS COURANTES DE DEUX VARIABLES

4.1. Fonction NI, ou NON-OU, ou NOR (Not OR)

C'est la fonction OU inversée. Elle se note :

$f(x,y) = \overline{x + y}$ (prononcer x ou y barre)

Le tableau de la figure 1.5 donne ses caractéristiques.

variables		Fonction OU	Fonction NI (NOR)
x	y	$x+y$	$\overline{x+y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Fig. 1.5. — *Tableau de vérité de la fonction NI ou NOR.*

4.2. Fonction NON-ET ou NAND (Not AND)

C'est la fonction ET inversée. Elle se note :

$f(x,y) = \overline{x.y}$ (prononcer x et y barre)

Le tableau de la figure 1.6 définit cette fonction.

variables		Fonction ET	Fonction NON ET (NAND)
x	y	xy	$\overline{xy}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Fig. 1.6. — *Tableau de vérité de la fonction NON-ET (NAND).*

4.3. Fonction OU exclusif

Cette fonction $f(x,y)$ de deux variables prend la valeur 1 seulement si une seule des variables vaut 1. La fonction OU exclusif se note :

$f(x,y) = x \oplus y$ (prononcer x ou exclusif y)

La figure 1.7 donne son tableau de vérité.

Variables		Fonction OU exclusif
x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 1.7. — *Tableau de vérité de la fonction OU exclusif.*

4.4. Fonction identité

Cette fonction $f(x,y)$ de deux variables prend la valeur 1 seulement si $x=y$. On note cette fonction :

$$f(x,y) = x \equiv y \qquad \text{(prononcer } x \text{ identique } y\text{)}$$

ou $f(x,y) = x \odot y$

Le tableau de la figure 1.8 donne les valeurs de cette fonction.

variables		Fonction identité
x	y	x⊙y
0	0	1
0	1	0
1	0	0
1	1	1

Fig. 1.8. — *Tableau de vérité de la fonction identité.*

On peut remarquer, en comparant les tableaux des figures 1.7 et 1.8 qu'à chaque combinaison de x et y, les fonctionx $x \oplus y$ et $x \odot y$ prennent des valeurs opposées :

$$x \oplus y = \overline{x \odot y}$$

4.5. Fonction logique quelconque complètement définie

Les fonctions logiques citées précédemment ont reçu des noms particuliers car elles interviennent fréquemment dans les schémas logiques. Elles ont été réalisées sous forme de circuits logiques que l'on trouvera dans des catalogues de constructeurs (NON, ET, OU, NAND, NOR...). Ces fonctions ne couvrent pas toutes les possibilités. Les autres sont alors des fonctions quelconques, sans nom particulier, définies seulement par un tableau de vérité qui donne leur valeur pour toutes les combinaisons des variables. La figure 1.9 donne le tableau de vérité d'une fonction quelconque de trois variables.

variables			Fonction
x	y	z	f(x, y, z)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Fig. 1.9. — *Tableau de vérité d'une fonction quelconque.*

4.6. Fonction logique quelconque incomplètement définie

Jusqu'à présent, nous avons étudié des fonctions qui prenaient une valeur 1 ou 0 pour chacune des combinaisons des variables. S'il y a n variables, il y a 2^n combinaisons de ces variables ($2^1 = 2$ pour une variable, $2^2 = 4$ pour deux variables, $2^3 = 8$ pour trois variables, etc...). Or il peut arriver que certaines de ces 2^n combinaisons soit impossible (dépendant par exemple du sens physique des variables) ou encore pour que certaines combinaisons des variables, la fonction f puisse prendre une valeur 0 ou 1 sans que cela ait d'importance pour le système. Pour ces combinaisons, on dit que la fonction f n'est pas définie, ou encore qu'elle est incomplètement définie. Dans le tableau de vérité, on note sa valeur par un X pour ces cas-là, ce qui signifie que f peut prendre la valeur 1 ou 0. La figure 1.10 en donne un exemple.

Variables			Fonction
x	y	z	f(x, y, z)
0	0	0	0
0	0	1	X
0	1	0	1
0	1	1	0
1	0	0	X
1	0	1	1
1	1	0	0
1	1	1	1

Fig. 1.10. — *Tableau de vérité d'une fonction quelconque incomplètement définie.*

5. PROPRIÉTÉS DES OPÉRATEURS ET, OU, NON

A partir des tableaux de vérité des fonctions ET, OU, NON, on peut démontrer quelques identités remarquables, qui sont ensuite très utiles lorsque l'on étudie des fonctions complexes. Nous donnons les résultats de ces démonstrations à la figure 1.11. Le lecteur pourra les retrouver pour les cas simples.

On constate que certaines identités se rapprochent de celles de l'algèbre classique :

$x + 0 = x \quad x.0 = 0$

$x(y + z) = xy + xz$

etc...

tandis que d'autres s'en écartent :

$x + 1 = 1$

$x.x = x$

$x + x = x$

$x + yz = (x + y)\,(x + z)$

D'autre part il existe des théorèmes généraux, et en particulier ceux de De Morgan et de Shannon, qui permettent la manipulations des fonctions logiques.

Théorème de De Morgan :

$\overline{x + y + z + \ldots} = \bar{x}\bar{y}\bar{z}\ldots$

$\overline{x.y.z.\ldots} \qquad = \bar{x} + \bar{y} + \bar{z} + \ldots$

Théorème de Shannon

$f(\overline{x,\ y,\ z,\ +,\ .}) = f(\bar{x},\ \bar{y},\ \bar{z},\ .,\ +)$

Ces théorèmes se vérifient avec une table de vérité.

Les théorème de De Morgan et de Shannon sont à appliquer avec précaution. Ils consistent à remplacer les variables par leurs inverses, les opérateurs OU par des opérateurs ET et vice versa. Soit à calculer l'inverse de la fonction $f = x + \bar{y}z$.

$\bar{f} = \overline{x + \bar{y}z}$ n'est pas $\bar{x}.y + \bar{z}$ mais

$\bar{f} = \overline{x + A} = \bar{x}.\,\bar{A} = \bar{x}.\,(\overline{\bar{y}z}) = \bar{x}(y + \bar{z}) = \bar{x}y + \bar{x}\bar{z}$

fonction différente de celle trouvée plus haut.

Fonctions	identités	observations
1 variable	$\bar{\bar{x}} = x$ $x+0=x \quad x.0=0$ $x+1=1 \quad x.1=x$ $x+x=x \quad x.x=x$ $x+\bar{x}=1 \quad x.\bar{x}=0$	
2 Variables	$x+y=y+x$ $x.y=y.x$	commutativité
3 variables	$x+y+z=(x+y)+z=x+(y+z)$ $x\ y\ z=(x\ y)z=x(y\ z)$	associativité
	$x(y+z)=x.y+xz$ $x+yz=(x+y)(x+z)$	distributivité
nombre quelconque de variables	$\overline{x+y+z+\ldots} = \bar{x}.\bar{y}.\bar{z}.\ldots$ $\overline{x.y.z.\ldots} = \bar{x}+\bar{y}+\bar{z}+\ldots$	Théorème de De Morgan
	$\overline{f(x,y,z,+,\cdot)} = f(\bar{x},\bar{y},\bar{z},\cdot,+)$	Théorème de Shannon
Fonction quelconque	Elle peut s'exprimer avec les seuls opérateurs ET, OU, NON.	

Fig. 1.11. — *Les propriétés des fonctions ET, OU, NON.*

Toutes les propriétés indiquées dans le tableau de la figure 1.11 sont utiles pour écrire les fonctions logiques quelconques et pour en simplifier l'écriture.

6. ÉCRITURES CANONIQUES D'UNE FONCTION LOGIQUE

Dans les paragraphes 2, 3 et 4, nous avons introduit les notations ., +, −, ⊕, ⊙ pour représenter les opérations ET, OU, NON, OU exclusif, Identité. Il existe également d'autres symboles moins courants pour les fonctions NAND, NOR. Dans la pratique, on utilise un nombre réduit de symboles pour écrire une fonction booléenne. Il s'agit surtout des symboles ET, OU, NON, car il a été démontré (résultat en figure 1.11) que ces trois symboles suffisaient à écrire toute fonction booléenne d'un nombre quelconque de variables, aussi complexe soit elle.

6.1. Fonctions produits

Soient trois variables binaires x, y, z. Pour chacune des huit combinaisons C_i (000, 001, 010, etc...), on définit une fonction P_i égale au ET des variables x ou $\bar{x}$, par y ou $\bar{y}$, et par z et $\bar{z}$, chacune des variables étant en direct si dans la combinaison C_i elle est à la valeur 1 ou en inverse si elle y figure à la valeur 0. Pour la combinaison $x = 0$, $y = 1$, $z = 0$, on obtient :

$x = 0$ valeur 0 variable inverse $\bar{x}$

$y = 1$ valeur 1 variable directe y

$z = 0$ valeur 0 variable inverse $\bar{z}$

d'où $P = \bar{x}.y.\bar{z}$

On établit ainsi huit fonctions P_0 à P_7 appelées *produit*. La figure 1.12 montre que ces fonctions ne valent 1 que pour la combinaison correspondante.

Cas	variables			Fonctions produits							
	x	y	z	$P_0 = \bar{x}\,\bar{y}\,\bar{z}$	$P_1 = \bar{x}\,\bar{y}\,z$	$P_2 = \bar{x}\,y\,\bar{z}$	$P_3 = \bar{x}\,y\,z$	$P_4 = x\,\bar{y}\,\bar{z}$	$P_5 = x\,\bar{y}\,z$	$P_6 = x\,y\,\bar{z}$	$P_7 = x\,y\,z$
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

Fig. 1.12. — *Fonctions produits.*

6.2. Fonctions quelconques écrites en somme de produits

Soit F une fonction quelconque de trois variables x, y et z. On peut écrire le tableau de vérité de F et lui associer les huit fonctions produits précédemment définies (fig. 1.13). On constate que F vaut 1 en même temps que P_1 pour le cas 1, que P_3 pour le cas 3 et P_4 pour le cas 4. Si l'on calcule la fonction :

$f = P_1 + P_3 + P_4$, on constate que $F = f$

Or les fonctions P_1, P_3, P_4 s'expriment en fonction de x, y, z comme il a été vu au paragraphe 6.1.

$$P_1 = \bar{x}\,\bar{y}\,z, P_3 = \bar{x}\,y\,z, P_4 = x\,\bar{y}\,\bar{z} \text{ d'où } F = \bar{x}\,\bar{y}\,z + \bar{x}\,y\,z + x\,\bar{y}\,\bar{z}$$

Cette façon d'écrire une fonction booléenne en réunissant par des OU *tous* les termes "produits" pour lesquels la fonction vaut 1 est appelée *somme canonique de produits.*

Cas	variables			F	Fonctions produits								$f = P_1 + P_3 + P_4$
	x	y	z		P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	
0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	[1]	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0	0	0	0	0	0
3	0	1	1	1	0	0	0	[1]	0	0	0	0	1
4	1	0	0	1	0	0	0	0	[1]	0	0	0	1
5	1	0	1	0	0	0	0	0	0	0	0	0	0
6	1	1	0	0	0	0	0	0	0	0	1	0	0
7	1	1	1	0	0	0	0	0	0	0	0	1	0

Fig. 1.13. — *Fonction F et produits correspondants.*

6.3. Fonctions sommes

Soient trois variables binaires x, y et z. Pour chacune des huit combinaisons C_i, on définit une fonction S_i égale au OU des variables x ou $\bar{x}$, y ou $\bar{y}$, z ou $\bar{z}$, chacune des variables étant prise en valeur directe si dans la combinaison C_i elle est à la valeur 0, ou en valeur inverse si elle y figure à la valeur 1. Pour la combinaison $x = 0$, $y = 1$, $z = 0$, on obtient :

$x = 0$ valeur 0 variable directe x

$y = 1$ valeur 1 variable inverse $\bar{y}$

$z = 0$ valeur 0 variable directe z

d'où $S = x + \bar{y} + z$,

On établit ainsi huit fonctions S_0 à S_7, appelées *sommes*. La figure 1.14 montre que ces fonctions ne valent 0 que pour la combinaison correspondante.

Cas	Variables			Fonctions sommes							
	x	y	z	$S_0 = x+y+z$	$S_1 = x+y+\bar{z}$	$S_2 = x+\bar{y}+z$	$S_3 = x+\bar{y}+\bar{z}$	$S_4 = \bar{x}+y+z$	$S_5 = \bar{x}+y+\bar{z}$	$S_6 = \bar{x}+\bar{y}+z$	$S_7 = \bar{x}+\bar{y}+\bar{z}$
0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	0	1	1	1	1	1	1
2	0	1	0	1	1	0	1	1	1	1	1
3	0	1	1	1	1	1	0	1	1	1	1
4	1	0	0	1	1	1	1	0	1	1	1
5	1	0	1	1	1	1	1	1	0	1	1
6	1	1	0	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	1	1	1	0

Fig. 1.14. — *Fonctions somme.*

6.4. Fonctions quelconques écrites en produit de sommes

Soit F la fonction quelconque de trois variables x, y et z utilisée au paragraphe 6.2. La figure 1.15 donne son tableau de vérité en lui associant les huit fonctions somme du paragraphe 6.3.

Cas	variables			F	Fonctions sommes								f =
	x	y	z		S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_0, S_2, S_5, S_6, S_7
0	0	0	0	0	[0]	1	1	1	1	1	1	1	0
1	0	0	1	1	1	0	1	1	1	1	1	1	1
2	0	1	0	0	1	1	[0]	1	1	1	1	1	0
3	0	1	1	1	1	1	1	0	1	1	1	1	1
4	1	0	0	1	1	1	1	1	0	1	1	1	1
5	1	0	1	0	1	1	1	1	1	[0]	1	1	0
6	1	1	0	0	1	1	1	1	1	1	[0]	1	0
7	1	1	1	0	1	1	1	1	1	1	1	[0]	0

Fig. 1.15. — *Fonction F et sommes correspondantes.*

On constate que F vaut 0 en même temps que S_0, S_2, S_5, S_6, S_7 pour les cas correspondants. Si l'on calcule la fonction $f = S_0 \cdot S_2 \cdot S_5 \cdot S_6 \cdot S_7$, on voit que $f = F$. Or les fonctions S_0, S_2, etc... s'expriment en fonction de x, y, z comme il a été vu au paragraphe 6.3.

$$S_0 = x + y + z,\ S_2 = x + \bar{y} + z,\ S_5 = \bar{x} + y + \bar{z}, S_6 = \bar{x} + \bar{y} + z,$$
$$S_7 = \bar{x} + \bar{y} + \bar{z}$$

d'où

$$F = (x + y + z)(x + \bar{y} + z)(\bar{x} + y + \bar{z})(\bar{x} + \bar{y} + z)(\bar{x} + \bar{y} + \bar{z})$$

Cette façon d'écrire une fonction booléenne en faisant le ET de tous les termes en OU pour lesquels la fonction vaut 0 est appelée *produit canonique de sommes.*

Des deux formes canoniques, c'est la somme de produits qui est la plus utilisée.

7. SIMPLIFICATION DE L'ÉCRITURE DES FONCTIONS LOGIQUES

En algèbre classique, simplifier une fonction, c'est rechercher une forme d'écriture plus condensée ou plus facile à calculer. Par exemple:

$$f(a,b) = \frac{a^2 - b^2}{a + b} \text{ devient } f(a,b) = a - b \text{ (si } a + b \neq 0)$$

En algèbre de Boole, simplifier l'écriture, c'est trouver une forme plus condensée, avec moins de symboles, qui conduit à une réalisation matérielle plus compacte. Les méthodes de simplification employées en algèbre de Boole utilisent les identités de la figure 1.11, et essentiellement:

$$x + \bar{x} = 1,\ x\bar{x} = 0,\ 1 + x = 1$$

De nombreux travaux ont été menés dans le domaine de la simplification. Nous exposerons deux méthodes: simplification par manipulation algébrique et par méthode de Karnaugh.

8. SIMPLIFICATION ALGÉBRIQUE

On souhaite simplifier la fonction f, définie par la table de vérité de la figure 1.16.

x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Fig. 1.16. — *Fonction f à simplifier.*

On obtient la forme canonique somme de produits comme indiqué au paragraphe 7 :

$$f = \bar{x}\, y\, z + x\, \bar{y}\, z + x\, y\, \bar{z} + x\, y\, z$$

On peut écrire une nouvelle forme de f :

$$f = (\bar{x}\, y\, z + x\, y\, z) + (x\, \bar{y}\, z + x\, y\, z) + (x\, y\, \bar{z} + x\, y\, z)$$

car $x\, y\, z + x\, y\, z + x\, y\, z = x\, y\, z$

$$f = yz\,(\bar{x} + x) + xz\,(\bar{y} + y) + xy\,(\bar{z} + z)$$

or $\bar{x} + x = 1$, $\bar{y} + y = 1$, $\bar{z} + z = 1$

d'où la forme simplifiée :

$$f = yz + xz + xy$$

La conclusion de tous ces calculs est que les procédés algébriques reposent beaucoup sur l'astuce et la chance de mettre en évidence les bons regroupements. De plus, au-delà de trois variables, les calculs algébriques deviennent très malaisés et la méthode ne s'applique pratiquement pas.

9. SIMPLIFICATION PAR LA MÉTHODE DE KARNAUGH

9.1. Construction des tableaux

La méthode de Karnaugh utilise l'identité $ax + a\bar{x} = a$. Elle consiste à mettre en évidence, par une méthode graphique tous les termes produits

d'une fonction qui ne diffèrent que par l'état d'une seule variable (termes dits adjacents). Ainsi les deux produits $\bar{x}\,y\,\bar{z}\,t$ et $\bar{x}\,y\,\bar{z}\,\bar{t}$ sont adjacents car ils ne diffèrent que par t. On a $\bar{x}\,y\,\bar{z}\,t + \bar{x}\,y\,\bar{z}\,\bar{t} = \bar{x}\,y\,\bar{z}\,(\bar{t} + t) = \bar{x}\,y\,\bar{z}$. On a supprimé ainsi la variable t et l'on a un seul produit au lieu de deux.

Si une fonction dépend de n variables, il y a au maximum 2^n produits. Chacun de ces 2^n produits est représenté par une case d'un tableau, construit de telle manière que les lignes soient adjacentes entre elles, et les colonnes entre elles.

Les figures 1.17 à 1.20 montrent les tableaux de Karnaugh pour 2, 3, 4, 5 et 6 variables.

Fig. 1.17. — *Deux formes d'un tableau de Karnaugh à deux variables x, y.*

Fig. 1.18. — *Deux formes d'un tableau de Karnaugh à trois variables x, y, z.*

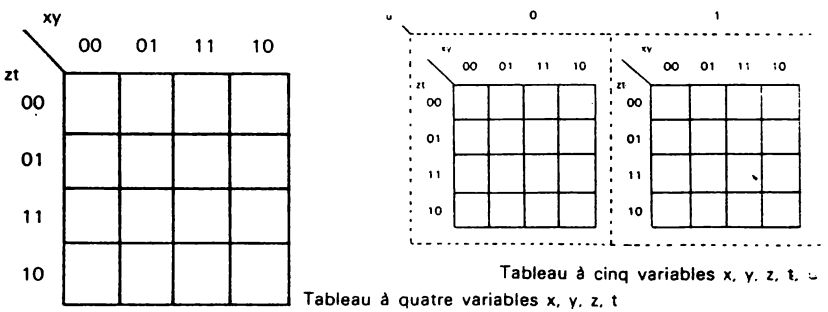


Fig. 1.19. — *Tableaux à quatre ou cinq variables.*

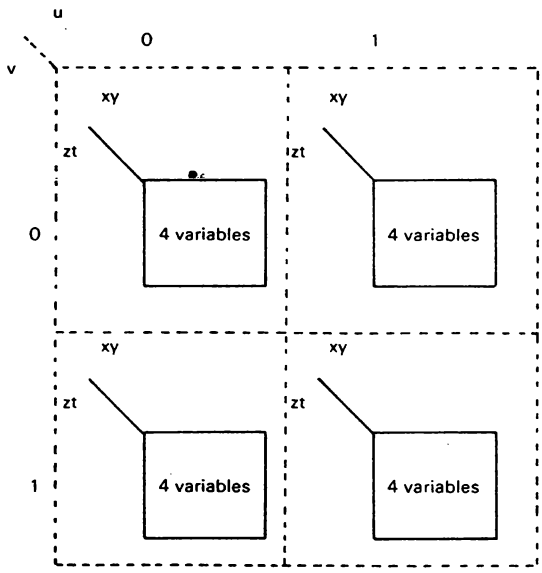


Fig. 1.20. — *Tableau à six variables x, y, z, t, u, v.*

Pour les tableaux à deux et trois variables l'expression des monômes a été reportée dans les cases. Deux types de tableaux apparaissent pour deux et trois variables: un type où les colonnes et lignes sont repérées par les valeurs 0 et 1 des variables et un autre type où les variables directes et inverses sont utilisées. Pour les tableaux suivants nous n'avons conservé que la forme avec les 0 et 1.

Le tableau à deux variables possède quatre cases, celui à trois variables huit cases; pour 4, 5, 6 variables on a respectivement 16, 32 et 64 cases. Les tableaux de Karnaugh à cinq et six variables sont formés par juxtaposition de tableaux à quatre variables.

Pour un tableau à quatre variables (x, y, z, t) on vérifie que l'on passe d'une colonne à la suivante en ne changeant qu'une seule variable (x ou y dans l'exemple) et que l'on passe d'une ligne à la suivante en ne changeant également qu'une seule variable (z ou t). La figure 1.21 montre les cases adjacentes (marquées d'une croix) et une case donnée (hachurée) pour un tableau à quatre variables.

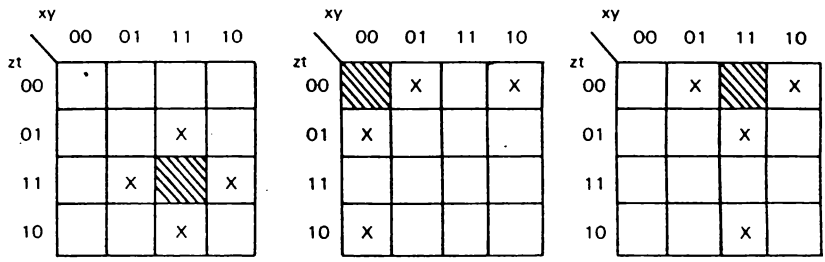


Fig. 1.21. — *Adjacences à une case donnée.*

9.2. Passage du tableau de vérité au tableau de Karnaugh

Le passage du tableau de vérité au tableau de Karnaugh consiste à écrire des 1 dans les cases correspondant aux combinaisons où la fonction vaut 1 et des zéros partout ailleurs (fig. 1.22):

xyzt	F
0000	1
0001	0
0010	1
0011	0
0100	0
0101	1
0110	0
0111	0
1000	1
1001	1
1010	1
1011	1
1100	0
1101	1
1110	0
1111	0

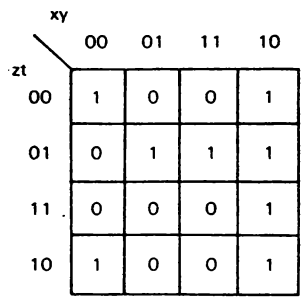


Fig. 1.22. — *Passage du tableau de vérité au tableau de Karnaugh.*

9.3. Méthode de Karnaugh

La méthode de Karnaugh est simple à mettre en œuvre:

- On transpose le tableau de vérité dans un tableau de Karnaugh.
- On réalise les groupements possibles de 1, 2, 4, 8 termes. On cherche à avoir le minimum de groupements, chaque groupement rassemblant le maximum de termes. On doit utiliser tous les termes. Les mêmes termes peuvent participer à plusieurs groupements (car $x + x + x = x$).
- Dans un groupement de un terme on ne peut éliminer aucune variable. On écrit donc le produit des variables directes et inverses caractérisant la case.
- Dans un groupement de deux termes on élimine la variable qui change d'état et l'on conserve le produit des variables directes ou inverses qui n'ont pas changé d'état dans le groupement (car $ax + a\bar{x} = a$).
- Dans un groupement de quatre termes on élimine les deux variables qui changent d'état et l'on conserve le produit des variables directes ou inverses qui n'ont pas changé d'état dans le groupement.
- etc..., on élimine trois variables dans un groupement de huit termes.
- L'expression logique finale est la réunion des groupements après élimination des variables ayant changé d'état.

9.4. Application à une fonction de quatre variables

La figure 1.23 donne quelques exemples de fonctions.

Pour la figure 1.23a on a un groupement de deux termes et un de quatre termes. Dans le groupement de deux termes x change d'état. Il reste y direct, z inverse et t direct soit $y\bar{z}t$. Dans le groupement de quatre termes, x et y changent d'état; il reste z direct et t inverse soit $z\bar{t}$. La fonction est donc la réunion des deux groupements après leur simplification:

$$f = y\bar{z}t + z\bar{t}$$

En traitant algébriquement les groupements indiqués à la figure 1.23a,

$$f = (\bar{x}y\bar{z}t + xy\bar{z}t) + (\bar{x}\bar{y}z\bar{t} + \bar{x}yz\bar{t} + xyz\bar{t} + x\bar{y}z\bar{t})$$

$$f = y\bar{z}t\,(\bar{x} + x) + z\bar{t}\,(\bar{x}\bar{y} + \bar{x}y + xy + x\bar{y})$$

$$f = y\bar{z}t + z\bar{t}\,(\bar{x}(\bar{y} + y) + x(y + \bar{y}))$$

$$f = y\bar{z}t + z\bar{t}(\bar{x}.1 + x.1)$$

$$f = y\bar{z}t + z\bar{t}\ (\bar{x} + x)$$

$$f = y\bar{z}t + z\bar{t}$$

On obtient évidemment le même résultat mais de manière plus pénible. Il faut de plus noter que le calcul algébrique a été facilité par les groupements suggérés par le tableau de Karnaugh.

La figure 1.23b montre deux groupements de quatre termes. Notons que les quatre coins du tableaux sont adjacents. On a, en appliquant les même procédés:

$$f = \bar{y}\bar{t} + yt$$

Dans la figure 1.23c on voit que l'on réutilise deux fois des termes produits. On obtient:

$$f = y + \bar{x}t$$

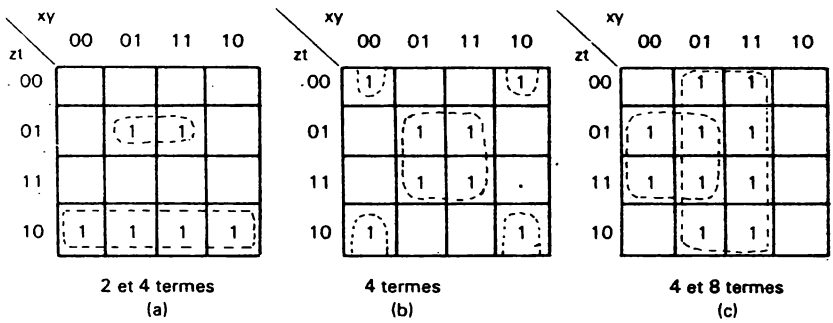
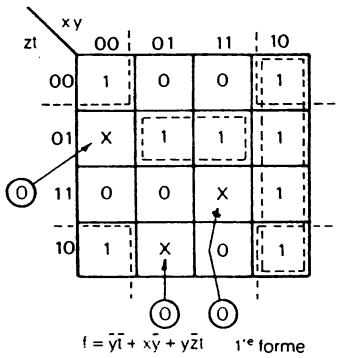
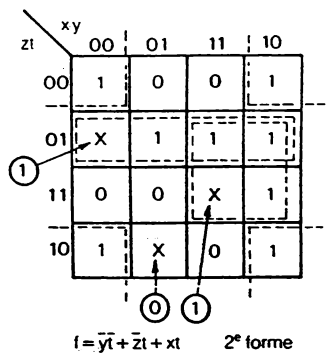


Fig. 1.23. — *Exemples de groupements adjacents.*

La figure 1.24 montre l'application de la méthode de Karnaugh pour une fonction incomplètement définie. Dans ce cas, on se sert des cases indéfinies en remplaçant les X par des 1 ou 0 selon que cela favorise les groupements de 1 ou non. On note que l'on peut réaliser des groupements différents qui conduisent à des formes simplifiées différentes. Cette pluralité des formes réduites existe aussi pour les fonctions complètement définies. Plus la fonction est complexe, plus le nombre de formes simplifiées peut être grand.

x	y	z	t	f
0	0	0	0	1
0	0	0	1	X
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	X
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	X





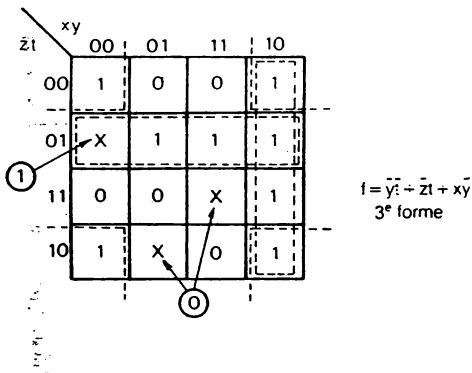


Fig. 1.24. — *Fonction incomplètement définie et tableaux de Karnaugh Diversité des groupements.*

9.5. Application à une fonction de cinq variables

Pour une fonction de cinq variables on recherche des groupements adjacents pris dans un tableau ou dans deux tableaux. Il faut alors que les groupements soient superposables dans les deux tableaux. La figure 1.25 donne un exemple:

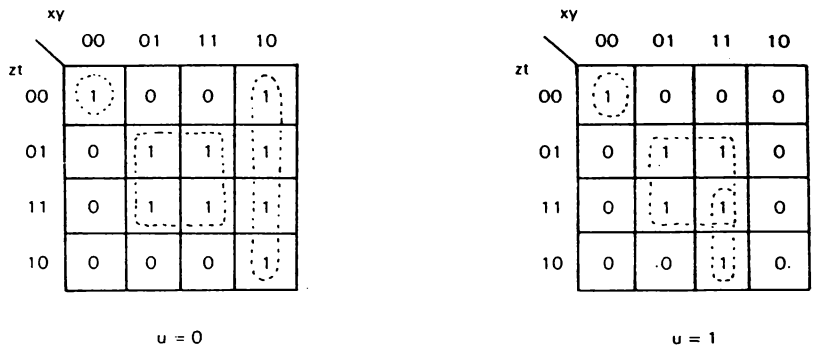


Fig. 1.25. — *Exemple de simplification pour cinq variables.*

On met en évidence :

- un groupement de une case sur les deux tableaux (coin gauche haut),
- un groupement de quatre cases sur les deux tableaux (milieu des deux tableaux),
- un groupement de quatre cases sur le tableau $\bar{u}$,
- un groupement de deux cases sur le tableau u.

En appliquant les règles on obtient :

$$f = \bar{x}\bar{y}\bar{z}\bar{t} + yt + x\bar{y}\bar{u} + xyzu$$

Le principe serait le même pour six variables. Il faudrait rechercher des groupements sur 1, 2 ou 4 tableaux. On atteint la limite pratique d'emploi de la méthode de Karnaugh.

10. MÉTHODES ALGORITHMIQUES

Des algorithmes de simplification des fonctions booléennes ont été développés pour des fonctions à plus de six variables (Méthode de Quine Mac Cluskey entre autres). Ces méthodes s'appliquent bien dans le cas où les fonctions sont peu nombreuses, où il y a peu de variables et qu'il existe beaucoup de monômes adjacents. En pratique, on rencontre rarement ce cas : les fonctions sont nombreuses, dépendent de beaucoup de variables et présentent peu des monômes adjacents. De plus, les circuits logiques actuels

permettent de réaliser économiquement des fonctions complexes, même non optimisées. Aussi pensons-nous que la méthode de Karnaugh suffit très largement pour traiter les cas de simplification rencontrés dans la pratique.*

11. SYMBOLES GRAPHIQUES ASSOCIÉS AUX FONCTIONS BOOLÉENNES

Comme nous le verrons au chapitre 4, les fonctions booléennes sont matérialisées par des circuits électriques, composés de résistances, diodes et transistors. Cette description électrique ne convient pas au logicien car celui-ci se préoccupe plutôt de savoir quelle fonction il obtient en sortie d'un opérateur en ayant appliqué des variables en entrée. Il utilise des schémas plus simples dont la forme graphique représente par convention le type de fonction réalisée :

- Fonction NON : un triangle suivi d'un rond,
- Fonction ET : un demi rond allongé,
- Fonction OU : un triangle curviligne,
- Etc...

Chacun de ces opérateurs s'appelle aussi porte. La figure 1.26 donne les symboles des portes utilisées couramment.

* L'optimisation des fonctions booléennes, un faux problème Automatisme n° 8-9, août-septembre 1977.

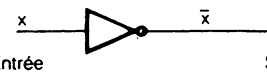
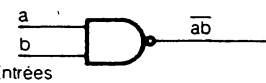
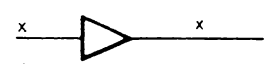
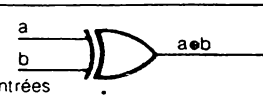
Fonction	Schéma symbolique (norme des catalogues)
Inverseur	
ET	
OU	
ET NON ou NAND	
OU NON ou NOR	
Séparateur (buffer)	
OU exclusif	

Fig. 1.26. — *Les symboles logiques courants.*

Les symboles de la figure 1.26 sont formés par l'association d'une forme et éventuellement d'un rond en sortie. On trouve également, dans des schémas de circuits complexes ou dans des assemblages de fonctions booléennes, des symboles avec des ronds en entrée, en entrée et en sortie. La figure 1.27 en montre quelques-uns. Ces symboles n'ont pas reçus de nom particulier au contraire des premiers. Pour déterminer la fonction réalisée par de tels opérateurs, on écrit :

- Que la fonction de sortie est directe s'il n'y a pas de rond en sortie de l'opérateur, et inverse s'il y a un rond en sortie.
- Que l'opérateur réalise la fonction prévue par la forme du symbolisme ET, OU.
- Que la fonction est réalisée sur des variables directes si les entrées n'ont pas de rond et, sur des variables inverses si elles ont un rond.

Ainsi, le symbole 1 de la figure 1.23 donne :

$$c = \bar{a} \cdot \bar{b}$$

et le symbole 3 :

$$d = \overline{\bar{a} \cdot \bar{b} \cdot c}$$

Sans rond, les entrées ou sorties sont dites *actives en direct*, avec rond, elles sont dites *actives en inverse.*

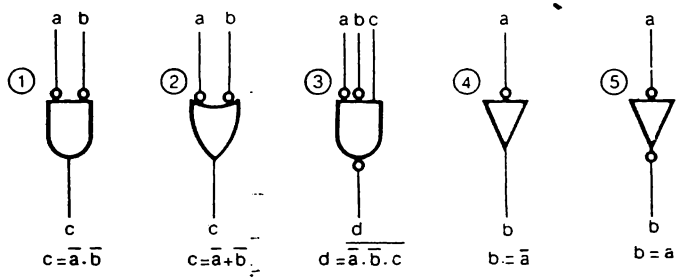


Fig. 1.27. — *Formes dérivées des symboles de base.*

12. LOGIGRAMME D'UNE FONCTION BOOLÉENNE

Le logigramme d'une fonction booléene est sa traduction en opérateurs de base ET, OU, NON, NAND, etc... Le logigramme dépend évidemment du choix des opérateurs. La figure 1.28 montre un logigramme de la fonction :

$$f = y \bar{z} t + x \bar{y} + \bar{y} \bar{t}$$

avec des ET, OU, NON.

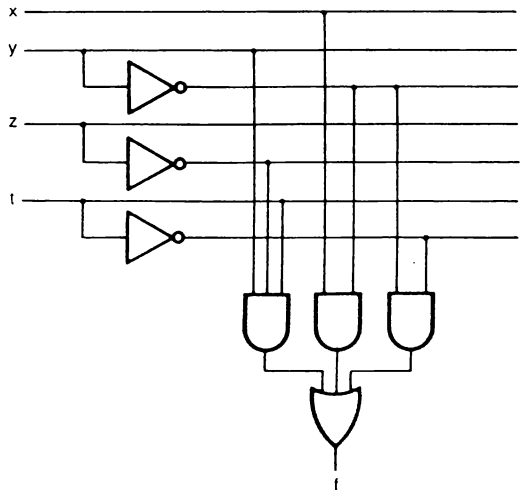


Fig. 1.28. — *Logigramme de la fonction f.*

Nous verrons dans le chapitre 9 qu'il y a plusieurs façons de réaliser des logigrammes de fonctions combinatoires et que la représentation en opérateurs élémentaires ne constitue pas toujours la meilleur solution.

Chapitre 2

CODAGE ET PROTECTION DE L'INFORMATION

1. DIVERS ASPECTS D'UNE MÊME INFORMATION

Nous nous intéressons dans ce chapitre à la représentation des nombres, des caractères alphabétiques, et à la protection de l'information contre les erreurs. Une même information peut se coder de plusieurs façons en fonction de son utilisation : par exemple, la forme d'un nombre change selon qu'il entre dans un bloc de calcul, qu'il apparaît sur un circuit d'affichage lumineux, sur un écran de visualisation, sur une imprimante, sur une carte perforée, etc...

2. REPRÉSENTATION DES NOMBRES ENTIERS

Il existe plusieurs façons de représenter des nombres :

- base 2 (binaire) avec deux symboles : 0 et 1,
- base 8 (octal) avec huit symboles : 0, 1, 2, 3, 4, 5, 6, 7,
- base 10 (décimal) avec dix symboles : 0 1 2 3 4 5 6 7 8 9

- base 16 (hexadécimal) avec seize symboles : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, les symboles lettres valant 10, 11, 12, 13, 14 et 15 en décimal,
- notation décimale codée binaire où chaque chiffre décimal de 0 à 9 est codé en binaire par 4 éléments binaires.

Dans les quatre premières représentations, dites pondérées un nombre N s'exprime à partir de la base B et des symboles a_i par la somme :

$$N = \sum_{i=0}^{n-1} a_i B^i$$

En pratique les symboles sont simplement juxtaposés, ceux de gauche représentant les poids forts et ceux de droite les poids faibles.

Exemples :

$N_2 = 10110 = 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0$
$= 16 + 4 + 2 = 22_{10}$

$N_8 = 572 = 5.8^2 + 7.8^1 + 2.8^0$
$= 5 \times 64 + 7 \times 8 + 2 \times 1 = 378_{10}$

$N_{10} = 782 = 7.10^2 + 8.10^1 + 2.10^0$
$= 700 + 80 + 2 = 782_{10}$

$N_{16} = 2A9F = 2.16^3 + 10.16^2 + 9.16^1 + 15.16^0$
$= 2 \times 4096 + 10 \times 256 + 9 \times 16 + 15 \times 1 = 10911_{10}$

$N_{DCB} = 1001\ 0010\ 1000 = 1001/0010/1000$
$= 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0/0.2^3 + 0.2^2 + 1.2^1 + 0.2^0/1.2^3$
$+ 0.2^2 + 0.2^1 + 0.2^0$
$= 9/2/8 = 928_{10}$

3. COMPTAGE DANS LES DIFFÉRENTS SYSTÈMES DE REPRÉSENTATION

La figure 2.1 montre la succession des nombres dans les différents systèmes de représentation :

Décimal	*Binaire*	*Octal*	*Hexadécimal*	*Décimal codé binaire*
00	000 000	00	00	0000 0000
01	000 001	01	01	0000 0001
02	000 010	02	02	0000 0010
03	000 011	03	03	0000 0011
04	000 100	04	04	0000 0100
05	000 101	05	05	0000 0101
06	000 110	06	06	0000 0110
07	000 111	07	07	0000 0111
08	001 000	10	08	0000 1000
09	001 001	11	09	0000 1001
10	001 010	12	0A	0001 0000
11	001 011	13	0B	0001 0001
12	001 100	14	0C	0001 0010
13	001 101	15	0D	0001 0011
14	001 110	16	0E	0001 0100
15	001 111	17	0F	0001 0101
16	010 000	20	10	0001 0110
17	010 001	21	11	0001 0111
18	010 010	22	12	0001 1000
19	010 011	23	13	0001 1001
20	010 100	24	14	0010 0000
21	010 101	25	15	0010 0001
22	010 110	26	16	0010 0010
23	010 111	27	17	0010 0011
24	011 000	30	18	0010 0100
25	011 001	31	19	0010 0101
26	011 010	32	1A	0010 0110
27	011 011	33	1B	0010 0111
28	011 100	34	1C	0010 1000
29	011 101	35	1D	0010 1001
30	011 110	36	1E	0011 0000
31	011 111	37	1F	0011 0001
32	100 000	40	20	0011 0010
33	100 001	41	21	0011 0011
34	100 010	42	22	0011 0100
35	100 011	43	23	0011 0101
36	100 100	44	24	0011 0110
37	100 101	45	25	0011 0111
38	100 110	46	26	0011 1000
39	100 111	47	27	0011 1001
40	101 000	50	28	0100 0000
41	101 001	51	29	0100 0001
42	101 010	52	2A	0100 0010
43	101 011	53	2B	0100 0011
44	101 100	54	2C	0100 0100
45	101 101	55	2D	0100 0101
46	101 110	56	2E	0100 0110
47	101 111	57	2F	0100 0111
48	110 000	60	30	0100 1000
49	110 001	61	31	0100 1001

Fig. 2.1. — *Nombres de 0 à 49 dans les différentes bases*

4. CHANGEMENTS DE BASE DE NUMÉROTATION

Soit un nombre N, exprimé dans une base B_1 à convertir dans une base B_2. On peut utiliser deux méthodes :

— soustractions successives de puissances de la base B_2,

— divisions successives du nombre N par la base B_2.

4.1. Soustractions successives

Examinons le cas du passage décimal au binaire :

$N_{10} = 1250$

On cherche la puissance de 2 immédiatement inférieure ou égale à 1250. C'est $2^{10} = 1024$. Donc :

$N = 1.2^{10} + 226$ $\quad (226 = 1250 - 1024)$

On essaie ensuite les puissances de 2 inférieures :

$2^9 = 512$. Donc :

$N = 1.2^{10} + 0.2^9 + 226$

$2^8 = 256$. Donc :

$N = 1.2^{10} + 0.2^9 + 0.2^8 + 226$

$2^7 = 128$. Donc :

$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 98$ $\quad (98 = 226 - 128)$

$2^6 = 64$. Donc :

$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 1.2^6 + 34$ $\quad (34 = 98 - 64)$

$2^5 = 32$. Donc :

$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 1.2^6 + 1.2^5 + 2$ $\quad (2 = 34 - 32)$

$2^4 = 16$. Donc :

$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 1.2^6 + 1.2^5 + 0.2^4 + 2$

$2^3 = 8$. Donc :

$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 1.2^6 + 1.2^5 + 0.2^4 + 0.2^3 + 2$

$2^2 = 4$. Donc :

$$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 1.2^6 + 1.2^5 + 0.2^4 + 0.2^3 + 0.2^2 + 2$$

$2^1 = 2$. Donc :

$$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 1.2^6 + 1.2^5 + 0.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 0 \qquad (0 = 2 - 2)$$

$$N = 1.2^{10} + 0.2^9 + 0.2^8 + 1.2^7 + 1.2^6 + 1.2^5 + 0.2^4 + 0.2^3 + 0.2^2 + 1.2^1 + 0.2^0$$

$$N_2 = 10011100010$$

4.2. Divisions successives

Soit un nombre N exprimé dans une base B_1, à convertir dans une base B_2. Dans la base B_2 le nombre N s'écrit :

$$N_{B_2} = a_{n-1} B_2^{n-1} + a_{n-2} B_2^{n-2} + \ldots a_1 B_2^1 + a_0$$

Si l'on divise N_{B_2} par B_2, on a la relation :

$$N_{B_2} = \underbrace{B_2 [a_{n-1} B^{n-2} + a_{n-2} B^{n-3} + \ldots a_1]}_{\text{quotient}} + \underbrace{a_0}_{\text{reste}}$$

Le reste de la première division donne le terme a_0. Si l'on divise le quotient à nouveau par B_2 ou obtient le terme a_1. Et ainsi de suite.

Appliquons cette méthode à la conversion binaire vers octal :

$$N_2 = a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \ldots a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0 2^0$$

$$N_2 = \underbrace{2^3 [a_{n-1} 2^{n-4} + a_{n-2} 2^{n-5} + \ldots a_5 2^2 + a_4 2^1 + a_3]}_{\text{quotient}} + \underbrace{a_2 2^2 + a_1 2^1 + a_0}_{\text{reste}}$$

Le reste représente le digit de poids faible du nombre en octal : il est constitué par les trois éléments binaires de poids faibles ($a_2\ a_1\ a_0$) du nombre exprimé en binaire.

Si l'on divise à nouveau le quotient par 8, on obtient comme reste les trois éléments binaires suivants ($a_5\ a_4\ a_3$) du nombre exprimé en binaire. Et ainsi de suite.

Pour passer d'un nombre exprimé en binaire à son équivalent octal, il

suffit de le découper en tranche de trois éléments binaires à partir de la droite et de coder ensuite chacune de ces tranches entre 0 et 7.

Exemple:

$$N_2 = 101/111/001 = 1.2^2 + 0.2^1 + 1.2^0/1.2^2 + 1.2^1 + 1.2^0/$$
$$0.2^2 + 0.2^1 + 1.2^0$$
$$= 5/7/1 = 571_8$$

Il est facile de démontrer que le passage du binaire où l'hexadécimal se fait en considérant des tranches de quatre chiffres binaires.

Exemple:

$$N_2 = 1011/0100/1100$$
$$= 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0/0.2^3 + 1.2^2 + 0.2^1 + 0.2^0/$$
$$1.2^3 + 1.2^2 + 0.2^1 + 0.2^0$$
$$= 11_{10}/4_{10}/12_{10} = B4C_{16}$$

Les transformations inverses, octal vers binaire ou hexadécimal vers binaire sont immédiates: on code en binaire sur trois ou quatre éléments binaires chacun des chiffres du nombre:

$$N_8 = 531_8 = 101\ 011\ 001_2$$
$$N_{16} = F5E_{16} = 1111\ 0101\ 1110_2$$

5. REPRÉSENTATION DES CARACTÈRES

5.1. Code ASCII

Dans ce qui précède, nous n'avons examiné que les nombres. Les machines traitent aussi des caractères alphabétiques (majuscules, minuscules), des signes de ponctuations, des caractères de délimitation. Il faut donc coder toutes ces entités pour que chaque caractère soit identifiable par une combinaison unique. Il existe de nombreux codes, dépendant des utilisations de l'information ou des supports de l'information. L'un des plus courants est le code CCITT n° 5 ou encore code ASCII. Il utilise 7 eb (éléments binaires) de telle sorte que l'on code:

- Les chiffres de 0 à 9.
- Les lettres majuscules et minuscules.

- Les signes de ponctuation.
- Des caractères de service utilisés pour commander les différents organes placés aux deux extrémités d'une ligne de transmission.
- Des caractères permettant de séparer les différents blocs d'information, etc...
- Des combinaisons de code disponibles laissées à la disposition des utilisateurs (NAT).

La figure 2.2 montre le tableau des valeurs du code ASCII.

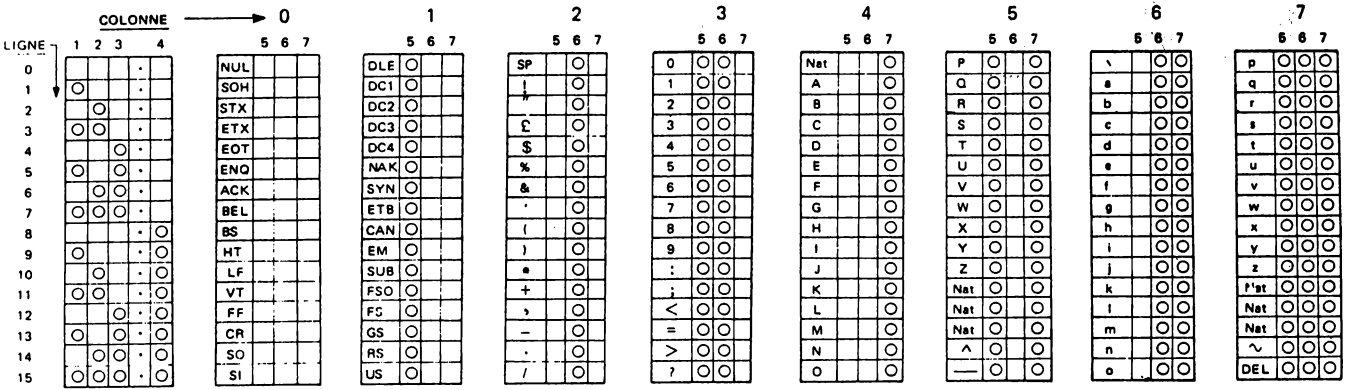
Les exemples suivants, tirés du tableau de la figure 2.2, illustrent quelques cas de codage :

- Chiffre 1 : eb 1234 567 = 1000 110
- Chiffre 9 : eb 1234 567 = 1001 110
- Lettre e : eb 1234 567 = 1010 011
- Signe + : eb 1234 567 = 1101 010

Pour les chiffres, les eb 1, 2, 3, 4 donnent la valeur numérique, 4 étant le poids fort et 1 le poids faible.

5.2. Autres codes

Selon les besoins particuliers d'un traitement, d'une transmission, ou les caractéristiques d'un support de stockage d'information, de nombreux codes de représentation ont été imaginés : code de Gray pour faire en sorte que des combinaisons adjacentes du code ne varient que par l'état d'une seule variable, afin d'éviter des changements multiples, code à excédent 3 pour certains organes de calcul, code des cartes perforées, code pour inscrire sur une bande magnétique à 7 ou 9 pistes, code pour imprimer ou visualiser un caractère par représentation matricielle de points, etc. Il serait assez fastidieux de les décrire tous. Certains codes sontégalement tombés en désuétude, les conditions de leur emploi ayant disparu.



SOH Début d'En-Tête
STX Début de Texte
ETX Fin de Texte
EOT Fin de Transmission
ENQ Demande
ACK Accusé de Réception
DLE Échappement Transmission
NAK Accusé de Réception Négatif
SYN Synchronisation
ETB Fin de Bloc de Transmission

BS Espace Arrière ←
HT Tabulation Horizontale
LF Interligne ↕
VT Tabulation Verticale
FF Présentation de Formule
CR Retour de Chariot ←
DC 1 Marche Lecteur
DC 2 Embrayage Perfo
DC 3 Arrêt Lecteur
DC 4 Débrayage Perfo

US Séparateur de Sous-Article
RS Séparateur d'Article
GS Séparateur de Groupe
FS Séparateur de Fichier

BEL Sonnerie
SO Hors Code
SI En Code
CAN Annulation
EM Fin de support
SUB Substitution
ESC Échappement
SP Espace →
NUL Nul
DEL Oblitération

Nat = Usage National

1 « Logique » (État Z)
0 « Logique » (État A)

Fig. 2.2. — *Code télégraphique n° 5 ou code ASCII.*

6. CHANGEMENT DE CODE

6.1. Principe

Si l'on dispose d'une quantité exprimée dans un code A, et que l'on veuille son équivalent dans un code B, il faut établir les fonctions B_1, B_2, B_3... en fonction des variables A_1, A_2, A_3... du code d'origine.

$B_1 = f_1\ (A_1, A_2, A_3, ...)$
$B_2 = f_2\ (A_1, A_2, A_3, ...)$
$B_3 = f_3\ (A_1, A_2, A_3, ...)$
etc.

6.2. Transcodage binaire vers décimal codé binaire

Soit un nombre binaire de 4 eb B_0, B_1, B_2, B_3 à transcoder en décimal codé binaire. Comme le nombre binaire est compris entre 0 et 15, il faut deux chiffres décimaux D_0 et D_1 chacun d'eux étant codés par 4 eb. La figure 2.3 donne le tableau de vérité.

	Code binaire				*Code décimal codé binaire*							
Décimal	b_3	b_2	b_1	b_0	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	1	0
3	0	0	1	1	0	0	0	0	0	0	1	1
4	0	1	0	0	0	0	0	0	0	1	0	0
5	0	1	0	1	0	0	0	0	0	1	0	1
6	0	1	1	0	0	0	0	0	0	1	1	0
7	0	1	1	1	0	0	0	0	0	1	1	1
8	1	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0	1
10	1	0	1	0	0	0	0	1	0	0	0	0
11	1	0	1	1	0	0	0	1	0	0	0	1
12	1	1	0	0	0	0	0	1	0	0	1	0
13	1	1	0	1	0	0	0	1	0	0	1	1
14	1	1	1	0	0	0	0	1	0	1	0	0
15	1	1	1	1	0	0	0	1	0	1	0	1

Fig. 2.3. — *Tableau de vérité pour le transcodage.*

On établit les tableaux de Karnaugh de d_0, d_1, d_2, d_3, d_4 en fonctions de b_0, b_1, b_2, b_3.

On constate que $d_0 = b_0$. Il reste à établir les tableaux de Karnaugh des quatre fonctions d_1, d_2, d_3, d_4 dépendant de b_0, b_1, b_2, b_3.

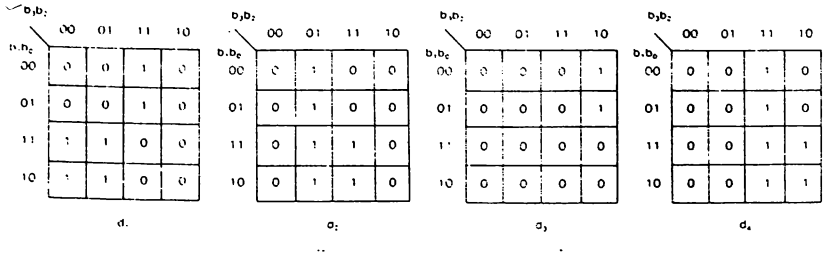


Fig. 2.4. — *Tableaux de Karnaugh du transcodage.*

On obtient :

$d_0 = b_0$

$d_1 = \overline{b}_3\ b_1 + b_3\ b_2\ \overline{b}_1$

$d_2 = \overline{b}_3\ b_2 + b_2\ b_1$

$d_3 = b_3\ \overline{b}_2\ \overline{b}_1$

$d_4 = b_3\ b_2 + b_3\ b_1$

7. LES CODES DÉTECTEURS D'ERREURS

Tout système de traitement ou de transmission de l'information manipule des nombres et des données codées en binaire en effectuant sur ceux-ci des opérations logiques ou arithmétiques, des transferts de données entre sous-ensembles sous forme parallèle ou série, etc. Si dans leur principe, toutes ces opérations sont parfaitement rigoureuses, leur réalisation, liée à la fiabilité des composants, aux aléas temporels de fonctionnement, aux variations de tension d'alimentation, etc. peut connaître des défauts.

Par exemple, pour la transmission de messages télé-informatiques, on souhaite avoir moins d'une erreur pour un million de bits transmis. Dans ce

cas les codes détecteurs sont calculés sur des blocs pouvant comporter plusieurs milliers d'éléments binaires.

Aussi, afin de se prémunir contre ces défauts, les théoriciens ont imaginé des modes de codage de l'information permettant de restaurer une information correcte malgré les aléas cités plus haut, ou plus modestement avertissant l'utilisateur que l'information est entachée d'erreur sans toutefois pouvoir la localiser.

Nous examinerons trois modes de codage pour illustrer notre propos.

7.1. Contrôle de parité ou d'imparité

7.1.1. Principe

Si dans un mode de codage déterminé, toutes les combinaisons sont utilisées pour représenter les informations, il n'est plus possible de distinguer une information erronée d'une information valable sur le simple examen du résultat car l'information erronée appartient au code. D'où l'idée de ne pas utiliser toutes les combinaisons possibles pour le codage des informations pour réserver des combinaisons à la détection des erreurs.

Le contrôle de parité consiste à ajouter un élément binaire dit de parité tel que le nombre de 1 dans l'information totale (utile + parité) soit pair.

Pour un contrôle d'imparité, on garde le même principe mais le nombre total de 1 doit être impair.

L'information utile peut se coder d'une manière quelconque. La figure 2.5 donne un exemple :

	Information totale		Quantité
	Élément de contrôle	Information utile	
eb de parité	0	1 0 1 0 1 1	A
	1	1 0 1 0 0 1	B
eb d'imparité	1	1 0 1 0 1 1	A
	0	1 0 1 0 0 1	B

Fig. 2.5. — *Parité ou imparité d'une information.*

7.1.2. Calcul de la parité ou de l'imparité

La figure 2.6 rappelle le tableau de vérité de la fonction OU exclusif (ou somme modulo 2) de deux variables *a* et *b*. On voit que cette fonction vaut 0 s'il y a un nombre pair de 1 et vaut 1 si il y a un nombre impair de 1. Le tableau de la fonction OU exclusif coïncide avec celui de la fonction parité.

$$\text{Parité}\,(a,b) = a \oplus b$$
$$\textit{Imparité}\,(a,b) = \overline{a \oplus b}$$

a	b	a ⊕ b	Parité	Imparité
0	0	0	0	1
0	1	1	1	0
1	0	1	1	0
1	1	0	0	1

Fig. 2.6. — *Tableau de vérité de la fonction OU exclusif de deux variables a et b.*

Par extension, on appelle OU exclusif de trois variables *a*, *b*, *c* la fonction $a \oplus b \oplus c$ telle que $a \oplus b \oplus c = (a \oplus b) \oplus c$. La figure 2.7 donne le tableau de vérité de cette fonction :

a	b	a ⊙ b	c	(a ⊙ b) ⊙ c	Parité a b c	Imparité a b c
0	0	0	0	0	0	1
0	0	0	1	1	1	0
0	1	1	0	1	1	0
0	1	1	1	0	0	1
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	1	0	0	0	0	1
1	1	0	1	1	1	0

Fig. 2.7. — *OU exclusif de trois variables, Parité et Imparité.*

On voit que la parité coïncide avec la fonction OU exclusif et l'imparité avec la fonction OU exclusif inversée. Ceci s'étend à un nombre quelconque de variables.

On peut vérifier sur le tableau de la figure 2.7 que l'inversion d'un nombre impair d'eb inverse les fonctions de parité et d'imparité, et l'inversion d'un nombre pair d'eb les laisse inchangées. Ceci montre les limites de ce mode de protection.

7.2. Codes *p* parmi *n*

La protection par parité ou imparité ne résiste pas à un changement simultané de deux variables par suite d'une erreur. Pour accroître la protection on représente le nombre par un plus grand nombre d'éléments binaires, de sorte qu'une combinaison erronée ait une plus grande chance de se trouver en dehors du code utile. La figure 2.8 représent un code dit 2 parmi 5 ou les dix chiffres de 0 à 9 sont codés par la présence de deux 1. Il y a donc 10 combinaisons utiles sur les 32 possibles. 22 combinaisons permettent de détecter des erreurs.

Code 2 parmi 5					*Valeur décimale*
A_4	A_3	A_2	A_1	A_0	
1	1	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	2
0	0	1	1	0	3
0	1	0	0	1	4
0	1	0	1	0	5
0	1	1	0	0	6
1	0	1	0	0	7
1	0	0	0	1	8
1	0	0	1	0	9

Fig. 2.8. — *Exemple de code 2 parmi 5.*

8. LES CODES CORRECTEURS D'ERREURS

En augmentant le nombre d'éléments binaires de contrôle, on peut créer des codes capables de déceler et même de corriger plusieurs erreurs.

8.1. Code de Hamming

8.1.1. Principe du code

Soit le code donné par le tableau de la figure 2.9, code à n éléments binaires dont h représente l'information utile et k les éléments de contrôle :

n							
k			h				valeur décimale
z	y	w	d	c	b	a	
1	2	3	4	5	6	7	
0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1
0	1	1	0	0	1	0	2
1	0	0	0	0	1	1	3
1	0	1	0	1	0	0	4
0	1	0	0	1	0	1	5
1	1	0	0	1	1	0	6
0	0	1	0	1	1	1	7
1	1	0	1	0	0	0	8
0	0	1	1	0	0	1	9
1	0	1	1	0	1	0	10
0	1	0	1	0	1	1	11
0	1	1	1	1	0	0	12
1	0	0	1	1	0	1	13
0	0	0	1	1	1	0	14
1	1	1	1	1	1	1	15

Fig. 2.9. — *Code de Hamming.*

Le code est construit de la façon suivante :

L'eb z est l'élément de parité des eb d'information d c a,

L'eb y est l'élément de parité des eb d'information d b a,

L'eb w est l'élément de parité des eb d'information c b a.

Le code de la figure 2.9 qui représente seulement 16 valeurs, utilise 7 eb, qui donnent 128 combinaisons. Il y a donc de grandes chances que l'altération d'une des 16 combinaisons du code fournisse une combinaison hors-code, détectable et même corrigeable.

8.1.2. Distance d'un code et possibilité de détection et correction

On appelle distance d'un code m, le nombre minimal de symboles binaires qui doivent être modifiés pour passer d'une combinaison du code à une autre combinaison du code, et ce pour toutes les combinaisons du code.

Dans le cas du code de la figure 2.9, la distance minimale est de 3. Pour s'en convaincre, il faut prendre chacune des combinaisons à 7 eb (*a*, *b*, *c*, *d*, *w*, *y*, *z*) et la comparer avec toutes les autres. On compte à chaque fois le nombre d'eb à inverser pour passer d'une combinaison à une autre et l'on garde le plus petit nombre. La distance minimale est la plus petite distance obtenue pour toutes les combinaisons.

Prenons par exemple le code 1 : 111 000 1

Sa distance aux codes 0 : 000 0000 est 4
2 : 011 0010 est 3
3 : 100 0011 est 3
4 : 101 0100 est 3

et ainsi de suite. Pour notre exemple, il faut effectuer 120 comparaisons pour obtenir la distance minimale.

Intuitivement, on "sent" que plus la distance d'un code est grande et moins il y a de combinaisons utilisées dans le code. Il en résulte qu'il y a alors beaucoup de combinaisons hors code qui permettent de détecter ou de corriger les erreurs.

On démontre que $m - 1 = c + d$ avec $c \leqslant d$

m : Distance minimale du code

c : Nombre d'erreurs pouvant être corrigées

d : Nombre d'erreurs pouvant être détectées.

Dans le cas de la figure 2.9 pour laquelle $m = 3$, il vient :

$$c + d = 2$$

Soit $c = 0$ et $d = 2$ ou $c = 1$ et $d = 1$.

La distance m sera d'autant plus grande, et donc les possibilités de détection et correction, que le nombre k d'éléments binaires de contrôle sera élevé par rapport au nombre h d'éléments binaires utiles.

8.1.3. Utilisation

A l'émission du message, on calcule les éléments de parités w, y et z, à partir des éléments de l'information h. Ils constituent la partie k du code n. A la réception du message n constitué de $h + k$, on calcule les sommes modulo 2 suivantes :

$$S_1 = z \oplus d \oplus c \oplus a$$
$$S_2 = y \oplus d \oplus b \oplus a$$
$$S_3 = w \oplus c \oplus b \oplus a$$

On appelle syndrome l'expression $S_3S_2S_1$. S'il n'y a aucune erreur dans la transmission du message le syndrome est nul puisqu'à l'émission, on avait calculé :

$$z = d \oplus c \oplus a,\ y = d \oplus b \oplus a,\ w = c \oplus b \oplus a$$

or $z \oplus (d \oplus c \oplus a) = z \oplus z = 0$
$y \oplus (d \oplus b \oplus a) = y \oplus y = 0$
$w \oplus (c \oplus b \oplus a) = w \oplus w = 0$

S'il y a des erreurs dans le message reçu, le syndrome prend une valeur différente de 0.

Supposons que l'erreur la plus vraisemblable soit l'inversion d'un seul eb de l'information totale $(h + k)$, le tableau de la figure 2.10 donne le résultat du calcul du syndrome pour les différentes inversions possibles d'un seul eb du code n :

eb changeant d'état	S_3	S_2	S_1
Aucun	0	0	0
a	1	1	1
b	1	1	0
c	1	0	1
d	0	1	1
w	1	0	0
y	0	1	0
z	0	0	1

Fig. 2.10. — *Calcul du syndrome.*

Pour une seule inversion d'eb, le syndrome prend une seule valeur. Le nombre $S_3\ S_2\ S_1$ permet donc de déterminer l'élément binaire que l'on doit inverser dans le nombre n pour corriger l'erreur.

Supposons maintenant que l'erreur la plus vraisemblable affecte un ou deux eb. Comme la distance du code est de 3, on obtient une combinaison hors code.

D'après la formule : $m - 1 = c + d$ avec $c \leqslant d$.

Si $m = 3$, $c = 0$, $d = 2$ et $c = 1$, $d = 1$.

On sait détecter deux erreurs mais sans les corriger ou en détecter une et la corriger. En tout état de cause, on ne sait pas corriger deux erreurs pour $m = 3$.

Pour s'en convaincre une fois de plus, on peut dresser un tableau donnant le syndrome $S_3\ S_2\ S_1$ pour l'inversion des couples d'éléments binaires *ab*, *ac*, *ad*, *aw*, etc... et l'on voit que des couples différents donnent des syndromes identiques. La valeur du syndrome ne permet donc plus de remonter aux eb en faute. Il y a en effet sept combinaisons $S_3\ S_2\ S_1$ différentes de 0 pour 28 erreurs doubles et simples.

En pratique, il existe des circuits calculant des codes de détection et correction d'erreur utilisant des propriétés voisines du code de Hamming (appelé code de Hamming modifié). Il faut ajouter :

- 5 eb de contrôle pour 8 eb utiles,
- 6 eb de contrôle pour 16 eb utiles,
- 7 eb de contrôle pour 32 eb utiles,
- 8 eb de contrôle pour 64 eb utiles.

Ces circuits permettent la correction d'une erreur et la détection de deux erreurs au plus. Dans certaines conditions, si les erreurs sont permanentes, (un circuit fournit toujours un "1" ou toujours un "0" quelle que soit l'information reçue), on peut corriger une erreur double avec les eb de contrôle destinés à une correction d'erreur simple.

Pour corriger une erreur double, un constructeur (National Semiconductor) propose de calculer les codes de Hamming modifiés sur l'information et l'information décalée. En comparant les deux syndromes ainsi obtenus, on peut déterminer le couple d'eb en erreur. Mais cela se paie cher. Il faut doubler les eb de contrôle nécessaires à la correction d'une simple erreur.

8.2. Parité croisée

Il est possible de corriger des erreurs dans une suite de combinaisons binaires en associant un contrôle de parité portant sur chaque combinaison avec un second contrôle de parité portant sur les éléments de même rang de diverses combinaisons.

	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1
	Information							P_1
	0	1	0	1	0	1	1	0
	1	0	1	0	1	1	1	1
	0	0	1	0	1	0	1	1
	0	0	0	0	0	0	0	0
	0	0	1	0	1	0	0	0
	1	1	1	1	0	0	1	1
	0	1	0	1	1	1	1	1
P_2	0	1	0	1	0	1	1	0
	0	0	1	0	1	1	0	1
	0	1	1	0	0	1	1	0
	0	1	1	0	1	0	1	0
	1	0	0	0	0	0	0	1
	1	1	1	1	0	0	0	0
	1	0	1	0	1	0	1	0
	1	0	0	0	1	0	1	1
P_2	0	1	1	1	0	0	0	1

Fig. 2.11. — *Parités transversales et longitudinales.*

Le tableau de la figure 2.11 représente une information codée pour chaque combinaison avec un élément P_1 de parité et pour chaque colonne avec un élément de parité P_2, toutes les sept combinaisons. Une erreur unique affectant une ligne et une colonne pourra être localisée. Une erreur double affectant une ligne ou une colonne sera détectable mais non localisable. Plusieurs erreurs peuvent être détectées à condition qu'une seule erreur se manifeste par ligne ou par colonne.

Chapitre 3

CODAGE DES NOMBRES ET OPÉRATIONS ARITHMÉTIQUES

1. PRINCIPE DES OPÉRATIONS ARITHMÉTIQUES

Les opérations d'addition, de soustraction, de multiplication et de division s'effectuent sans difficulté particulière. Les tables de la figure 3.1 en donnent les principes appliqués aux deux chiffres de base 0 et 1. La figure 3.2 montre la réalisation des quatres opérations de base sur des nombres binaires non signés.

Addition	0 + 0 = 0	0 + 1 = 1	1 + 0 = 1	1 + 1 = 10 (↑ retenue)
Soustraction	0 − 0 = 0	0 − 1 = 1 1 (↑ retenue)	1 − 0 = 1	1 − 1 = 0
Multiplication	0 × 0 = 0	0 × 1 = 0	1 × 0 = 0	1 × 1 = 1
Division	0 : 0 = X	1 : 0 = X	0 : 1 = 0	1 : 1 = 1
		résultat non défini		

Fig. 3.1. — *Tables des opérations en arithmétique binaire.*

- Addition (1101 + 111)

```
 1111   ← retenues
  1101
+  111
------
 10100
```

- Soustraction (1101 − 111)

```
  1101
−  111
------
   11   ← retenues
------
  0110
```

- Multiplication (1011 x 101)

```
   1011
 x  101
------
   1011
  0000
 1011
------
110111
```

- Division (10101 : 11)

```
10101 | 11
  100 | ---
    11 | 111
     0 |
```

Fig. 3.2. — *Les quatre opérations sur des nombres binaires non signés.*

2. LES DIFFÈRENTS MODES DE REPRÉSENTATION DES NOMBRES

Il existe plusieurs modes de représentation des nombres selon qu'ils sont non signés, signés, fractionnaires, etc. Nous étudions dans ce paragraphe les représentations les plus courantes.

2.1. Nombres non signés

Avec n eb, on représente des nombres compris entre 0 et $2^n - 1$ (fig. 3.3) :

Valeur décimale	Valeur binaire
7	1 1 1
6	1 1 0
5	1 0 1
4	1 0 0
3	0 1 1
2	0 1 0
1	0 0 1
0	0 0 0

Fig. 3.3. — *Nombres non signés avec 3 eb.*

2.2. Nombres signés sous la forme ± N (sign-magnitude)

L'idée la plus naturelle consiste à représenter les nombres comme en algèbre classique, avec un signe + ou — et une valeur absolue. Un nombre binaire comprend alors un eb de signe (0 pour les nombres positifs, et 1 pour les nombres négatifs par exemple) et des eb de valeurs. Avec n eb, on note les nombres de 0 à $2^{n-1} - 1$ et 0 à $-(2^{n-1} - 1)$. Le zéro possède deux représentations distinctes + 0 et — 0.

Nombre binaire		équivalence décimale
signe	valeur	
0	1 1	+ 3
0	1 0	+ 2
0	0 1	+ 1
0	0 0	+ 0
1	0 0	- 0
1	0 1	- 1
1	1 0	- 2
1	1 1	- 3

Fig. 3.4. — *Représentation des nombres sous la forme* $\pm |N|$

2.3. Nombres signés en complément à 2

2.3.1. Complément à 1 d'un nombre

Soit A un nombre à n eb:

$A = a_{n-1}\ a_{n-2} \cdots a_1\ a_0$

On appelle complément à 1 ou complément restreint de A le nombre A' tel que:

$A + A' = 2^n - 1$ (+ signifiant addition et non union).

A' s'obtient en inversant tous les eb de A. On vérifie en effet que $A + A' = 2^n - 1$.

$$\begin{array}{lll} & a_{n-1}\ a_{n-2} \cdots a_1\ a_0 & A \\ + & \overline{a}_{n-1}\ \overline{a}_{n-2} \cdots \overline{a}_1\ \overline{a}_0 & A' \\ \hline & 1 \quad 1 \qquad 1 \ 1 & \end{array}$$

2.3.2. Complément à 2 d'un nombre

Soit A un nombre à n eb

$A = a_{n-1}\ a_{n-2} \cdots a_1\ a_0$

On appelle complément à 2, ou complément vrai de A le nombre A'' tel que :

$$A + A'' = 2^n$$

comme $A + A' = 2^n - 1$, il vient :

$$A'' = A' + 1.$$

Ce que l'on vérifie ci-dessous :

$$\begin{array}{llllll} a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 & A \\ \overline{a}_{n-1} & \overline{a}_{n-2} & \cdots & \overline{a}_1 & \overline{a}_0 & A' \\ & & & & 1 & 1 \end{array} \Big\} A''$$

$$2^n \quad 1 \quad 0 \quad 0 \quad \ldots \quad 0 \quad 0$$

Comme le bloc de calcul traite des nombres limités à n eb, $A + A'' = 2^n$ est identique à $A + A'' = 0$ à 2^n près, d'où : $A'' = -A$. Le complément à 2 d'un nombre peut donc être considéré comme son opposé. On utilise cette propriété dans la représentation en complément à 2.

2.3.3. Représentation des nombres en complément à 2

Un nombre comprend n eb. Pour un nombre positif, les $n - 1$ eb de poïds faible représentent la valeur sous forme de somme pondérée comme pour un nombre sans signe (voir § 2.1). Le n^{ieme} eb, de poids fort est à 0 et représente le signe +. Par définition, le nombre négatif opposé d'un nombre

Valeur binaire			équivalence décimale
signe	valeur		
0	1	1	+ 3
0	1	0	+ 2
0	0	1	+ 1
0	0	0	+ 0
1	1	1	- 1
1	1	0	- 2
1	0	1	- 3
1	0	0	- 4

Fig. 3.5. — *Représentation des nombres sous la forme du complément à 2.*

positif donné, s'obtient en prenant le complément à 2 du nombre positif, signe compris. Avec n eb, on peut représenter des nombres positifs de 0 à $2^{n-1}-1$ et des nombres négatifs de -1 à -2^{n-1}. Les nombres positifs se caractérisent par un eb de signe à 0, et les nombres négatifs par un eb de signe à 1 (fig. 3.5). On notera la différence avec le tableau de la figure 3.4.

Vérifions les règles de passage:

$+1$ à -1: 001 → 110 + 1 → 111

-3 à $+3$: 101 → 010 + 1 → 011

Le plus petit des nombres négatifs est son propre opposé.

-4 à -4: 100 → 011 + 1 → 100.

Le zéro possède une seule représentation:

$+0 = 000$ $\qquad -0 = 111 + 1 = 000.$

2.4. Nombres fractionnaires en virgule fixe

Un nombre de n eb est partagé en deux parties dont l'une correspond aux valeurs entières, l'autre aux valeurs fractionnaires. La virgule n'est pas matérialisée réellement comme le signe, mais se trouve fictivement entre les deux zones entières et fractionnaires. La position de la virgule est fixée par l'utilisateur en fonction de ses besoins. Le tableau de la figure 3.6 montre une suite de valeurs en complément à 2 avec une zone entière de 3 eb et une zone fractionnaire de 2 eb.

Ainsi le nombre 011010 vaut:

$1.2^2 + 1.2^1 + 0.2^0 + 1.2^{-1} + 0.2^{-2}$

$4 + 2 + 0{,}5 = 6{,}50$

Le nombre 111101, qui est négatif, correspond au nombre positif $000010 + 1 = 000011$, soit:

$0.2^2 + 0.2^1 + 0.2^0 + 1.2^{-1} + 1.2^{-2} = 0{,}75$

111101 correspond donc à $-0{,}75$.

Le tableau de la figure 3.6 est tout-à-fait semblable à celui de la figure 3.5. C'est par convention que la virgule est introduite et qu'une fraction des eb correspond à des puissances négatives de 2.

Valeur binaire						Valeur décimale
Signe	Valeur entière			Valeur Fractionnaire		
0	1	1	1	1	1	+ 7,75
0	1	1	1	1	0	+ 7,50
0	1	1	1	0	1	+ 7,25
0	1	1	1	0	0	+ 7,00
0	1	1	0	1	1	+ 6,75
		Etc...				
0	0	0	0	0	1	+ 0,25
0	0	0	0	0	0	0,00
1	1	1	1	1	1	− 0,25
1	1	1	1	1	0	− 0,50
1	1	1	1	0	1	− 0,75
		Etc...				
1	0	0	0	0	1	− 7,75
1	0	0	0	0	0	− 8,00

Fig. 3.6. — *Représentation des nombres fractionnaires en complément à 2.*

2.5. Nombres en virgule flottante

La virgule fixe permet de représenter tous les nombres de la plage avec une même précision, égale à celle de l'eb de poids le plus faible (± 0,25 dans l'exemple de la figure 3.6). Mais ce mode ne permet pas de manipuler des nombres très petits ou très grands, comme on en rencontre dans les calculs scientifiques. Dans ces cas, on utilise la notation exponentielle, où les nombres sont de la forme:

$$\pm M.2^E$$

M désigne la mantisse et E l'exposant. Cette notation permet de représenter des nombres très grands avec E positif ou très petits avec E négatif. La précision de représentation de ces nombres dépend du nombre d'eb de la mantisse.

La virgule est dite flottante parce que si l'on écrivait le nombre en notation pondérée habituelle, sa position dépendrait de la valeur de l'exposant (elle "flotterait" avec celui-ci). A nombre d'eb égal, un nombre exprimé en flottant couvre un domaine plus grand que celui obtenu avec la virgule fixe, mais avec une précision moindre.

La figure 3.7 montre le format des nombres flottants manipulés dans le circuit *Am* 9511 du constructeur AMD. L'eb 31 représente le signe de la mantisse (0 pour positif, 1 pour négatif), les eb 0 à 23 la valeur absolue de la mantisse, et les eb 24 à 30, l'exposant en complément à 2.

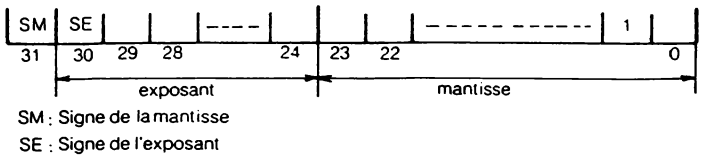


Fig. 3.7. — *Nombre en virgule flottante.*

L'eb 23 de la mantisse est toujours à 1. On dit alors que la mantisse est "normalisée". Elle représente un nombre fractionnaire compris entre:

0,100 ... 0 et 0,11 ... 1

soit entre 0,5 et $1 - 2^{-24}$, c'est-à-dire pratiquement 1. La valeur 0 n'est pas normalisable et se code par tous les eb de la mantisse à 0. L'exposant couvre la plage — 64 à + 63. Le domaine de représentation s'étend de -2^{63} à $+2^{63}$ ($\pm 9{,}2.10^{18}$). Les plus petits nombres exprimables sont $\pm 0{,}5.2^{-64}$ ($\pm 2{,}7.10^{-20}$).

La représentation donnée plus haut n'est pas unique pour les nombres en virgule flottante et dépend des circuits utilisés.

3. ADDITION ET SOUSTRACTION DE DEUX NOMBRES BINAIRES ENTIERS NON SIGNÉS

3.1. Principe de l'addition

Soit deux nombres A et B à quatre éléments binaires. La figure 3.8 montre comment obtenir leur somme.

La somme de a_1 et b_1 donne un résultat partiel S_1 et une retenue r_1. On forme alors la somme de r_1, a_2 et b_2 pour obtenir la somme partielle S_2 et la retenue r_2 et ainsi de suite.

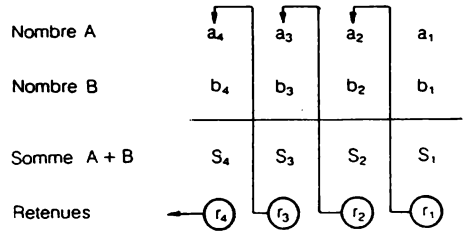


Fig. 3.8. — *Principe de l'addition.*

3.2. Principe de la soustraction

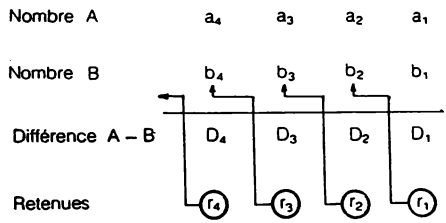


Fig. 3.9. — *Principe de la soustraction.*

La différence a_1 moins b_1 donne un résultat partiel D_1 et une retenue r_1. La somme $b_2 + r_1$ est soustraite de a_2 pour donner D_2 et la retenue r_2 et ainsi de suite (fig. 3.9).

3.3. Étude des conditions aux limites

Dans une machine numérique, les blocs de calcul ont toujours une taille limitée. Les nombres manipulables et les résultats obtenus ne comprennent alors que n eb, avec lesquels on représente des nombres compris entre 0 et $2^n - 1$. Additionner deux nombres compris entre 0 et $2^n - 1$ peut donc conduire à un résultat dépassant $2^n - 1$ (dépassement supérieur). De même, soustraire un nombre d'un nombre plus petit conduit à un nombre négatif (dépassement inférieur). Il faut donc s'accommoder de cette limitation et être prévenu des dépassements pour pouvoir interpréter correctement les résultats obtenus.

Examinons les différents cas :

— Si l'on additionne des nombres de telle façon que l'on ne dépasse pas $2^n - 1$, on obtient le résultat en vraie grandeur et une retenue terminale r_n à 0 (fig. 3.10).

— Si l'on soustrait un nombre d'un nombre plus grand, on obtient le résultat en vraie grandeur et une retenue terminale r_n à 0 (fig. 3.10).

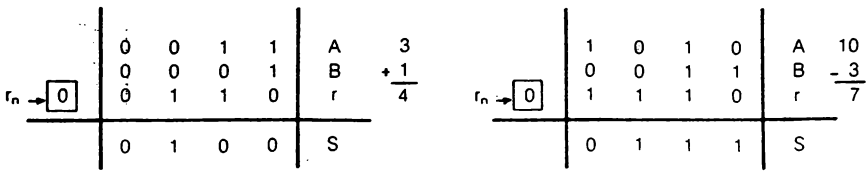


Fig. 3.10. — *Addition et soustraction en vraie grandeur.*

— Si l'on additionne des nombres tels que le résultat dépasse $2^n - 1$, on obtient une retenue terminale r_n à 1 et les poids faibles de la somme (fig. 3.11). On dit qu'il y a *dépassement.*

— Si l'on soustrait un nombre d'un autre nombre plus petit, on obtient une retenue terminale r_n à 1 et une différence qui n'est plus en vraie grandeur (fig. 3.11). Il y a également dépassement.

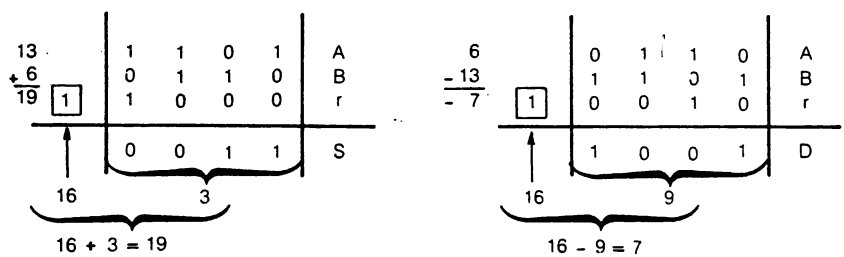


Fig. 3.11. — *Somme et différence avec dépassement.*

En conclusion, si après une addition ou une soustraction, on a une retenue terminale égale à 0, on obtient les résultats en vraie grandeur, et si l'on obtient une retenue terminale égale à 1, il y a dépassement.

3.4. Traitement du dépassement

Lorsqu'il y a un dépassement, on se trouve dans l'un ou l'autre des deux cas suivants :

- Le dépassement constitue une anomalie, le format de n eb devant en principe représenter toutes les valeurs possibles. Cela signifie qu'il y a eu une erreur dans la chaîne de traitement. On déclenche alors une alarme pour prévenir l'utilisateur.
- Le dépassement est normal, car par principe, le format à n eb est insuffisant pour représenter les valeurs possibles. On réalise alors les opérations en cascade sur plusieurs groupes de n eb. Si un dépassement se produit sur la dernière tranche, il est alors traité comme une anomalie.

La figure 3.12 montre à titre d'exemple une addition globale sur des nombres de 24 eb, avec un bloc de calcul à 24 eb.

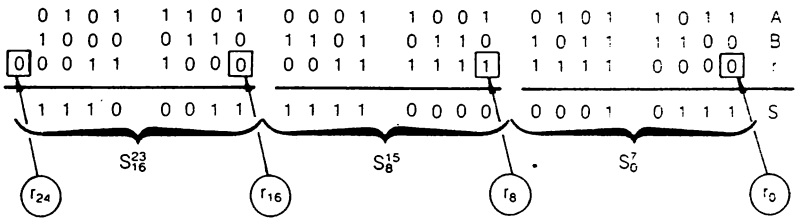


Fig. 3.12. — *Addition globale sur 24 eb.*

La même addition est réalisée en trois étapes avec un bloc de calcul à 8 eb (fig. 3.13). La retenue terminale de la première tranche, qui indique un dépassement sur la première tranche sert de retenue initiale à la deuxième. Il ne s'agit pas d'une anomalie dans le contexte d'une addition de nombres dépassant la capacité du bloc de calcul.

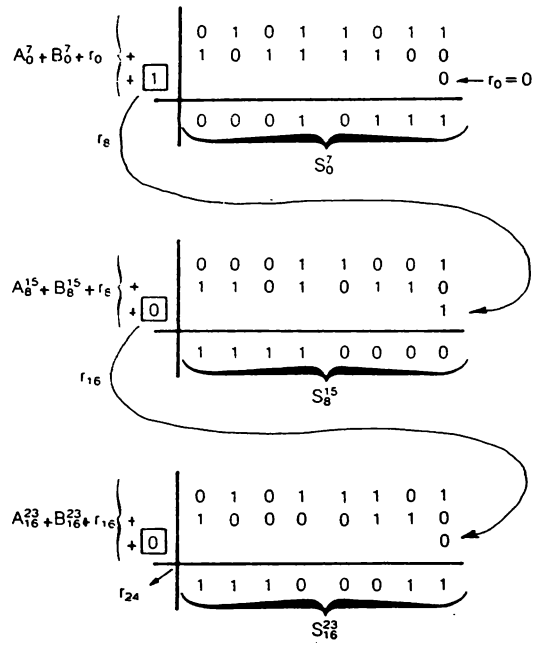


Fig. 3.13. — *Addition de 24 eb réalisée en 3 étapes.*

3.5. Soustraction ramenée à une addition

Dans le passé, lorsque les additionneurs étaient des circuits coûteux, on évitait de concevoir à la fois un additionneur et un soustracteur et l'on ramenait l'opération de soustraction à une opération d'addition, en utilisant les propriétés du complément à 2 :

$$A - B = A + (-B) = A + B''$$

car $-B$ *est représenté, à* 2^n près, par son complément à 2, B''.

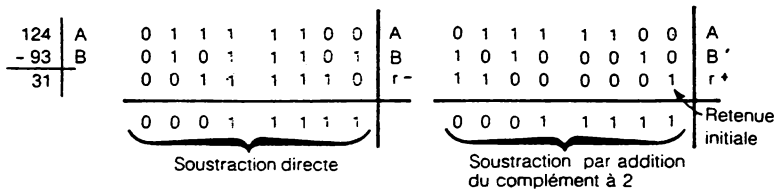


Fig. 3.14. — *Soustraction ramenée à une addition.*

Comme $B'' = B' + 1$, le complément vrai s'obtient simplement en inversant tous les eb de B et en ajoutant 1 sur le poids le plus faible. La figure 3.14 montre un exemple numérique.

Cette procédure était avantageuse lorsque l'on ne disposait que de circuits élémentaires. Elle ne présente plus le même intérêt aujourd'hui avec des circuits complexes où les opérations d'addition et de soustraction, ainsi que d'autres, sont réalisées dans le même circuit, sans que cela complique le montage du point de vue de l'utilisateur.

4. ADDITION ET SOUSTRACTION DE DEUX NOMBRES BINAIRES ENTIERS EN COMPLÉMENT A 2

4.1. Addition et soustraction

L'addition ou la soustraction pour des nombres représentés en mode algébrique avec signe et valeur absolue comme indiqué au paragraphe 2.2 n'est pas facile. Il faut tenir compte, pour donner un résultat correct du signe de chacun des deux nombres, du type d'opération (addition ou soustraction), et du classement relatif de leurs valeurs absolues. Par exemple:

$$3 + (-8) = 3 - 8 = -(8 - 3)$$

0 1 1	+ 7	
0 1 1 0	+ 6	
0 1 0 1	+ 5	
0 1 0 0	+ 4	
0 0 1 1	+ 3	
0 0 1 0	+ 2	
0 0 0 1	+ 1	
0 0 0 0	0	
1 1 1 1	- 1	
1 1 1 0	- 2	
1 1 0 1	- 3	
1 1 0 0	- 4	
1 0 1 1	- 5	
1 0 1 0	- 6	
1 0 0 1	- 7	
1 0 0 0	- 8	

+ 2	0 0 1 0	
+ + 3	+ 0 0 1 1	
+ 5	0 1 0 1	
+ 3	0 0 1 1	
+ - 2	+ 1 1 1 0	
+ 1	0 0 0 1	
- 3	1 1 0 1	
+ + 2	+ 0 0 1 0	
- 1	1 1 1 1	
- 3	1 1 0 1	
+ - 4	+ 1 1 0 0	
- 7	1 0 0 1	
+ 2	0 0 1 0	
- - 3	- 1 1 0 1	
+ 5	0 1 0 1	
- 7	1 0 0 1	
- - 3	- 1 1 0 1	
- 4	1 1 0 0	

Fig. 3.15. — *Différents exemples d'opérations en complément à 2.*

Dans ce cas, l'addition se ramène à une soustraction en inversant les opérandes. Le signe final dépend de la comparaison des valeurs absolues.

Le grand intérêt de la représentation en complément à 2 est que l'on ne s'occupe pas des signes. On additionne ou l'on soustrait les nombres, *eb de signe compris.* Le résultat est obtenu en complément à 2 avec le signe correct. Ceci se démontre mais est assez pénible. Prenons plutôt quelques exemples (fig. 3.15) pour illustrer les propriétés de ce mode de représentation.

4.2. Étude des conditions aux limites. Débordement.

Nous avions vu au paragraphe 3.3, en additionnant ou soustrayant des nombres non signés, qu'il pouvait y avoir dépassement de capacité, au-dessus de la valeur maximum $2^n - 1$ ou en-dessous de la valeur minimum 0. Il en est de même pour les nombres signés, les limites étant $2^{n-1} - 1$ pour les valeurs positives et -2^{n-1} pour les valeurs négatives.

Le tableau de vérité de la figure 3.16 donne les 16 cas possibles déterminés par les signes S_A et S_B des deux opérandes A et B, le type d'opération OP réalisé (0 pour l'addition, 1 pour la soustraction) et le signe S_R du résultat. Quand le signe S_R ne correspond pas à ce que l'on attend en appliquant les règles classiques de l'algèbre, on dit qu'il y a *débordement* ou overflow.

	S_A	S_B	OP	S_R	DEB
⓪	0	0	0	0	0
①	0	0	0	1	1
②	0	0	1	0	0
③	0	0	1	1	0
④	0	1	0	0	0
⑤	0	1	0	1	0
⑥	0	1	1	0	0
⑦	0	1	1	1	1
⑧	1	0	0	0	0
⑨	1	0	0	1	0
⑩	1	0	1	0	1
⑪	1	0	1	1	0
⑫	1	1	0	0	1
⑬	1	1	0	1	0
⑭	1	1	1	0	0
⑮	1	1	1	1	0

A, B } Opérandes
R Résultat de l'opération
S_A Signe de A, S_B Signe de B, S_R Signe de R } S = 0 si nombre positif, S = 1 si nombre négatif
OP Opérateur
OP = 0 Addition
OP = 1 Soustraction
DEB Débordement
DEB = 0 Pas de débordement
DEB = 1 Débordement

Fig. 3.16. — *Débordement en complément à 2 pour les opérations d'addition et de soustraction.*

Voyons d'abord les cas normaux :

Cas 0 : $A + B = R$ avec A, B, R positifs.

Cas 2 et 3 : $A - B = R$ avec A, B positifs et R positif ou négatif selon les valeurs de A et B.

Cas 4 et 5 : $A + B = R$ avec A positif, B négatif et R positif ou négatif selon les valeurs de A et B.

Cas 8 et 9 : $A + B = R$; semblables à 4 et 5.

Cas 11 : $A - B = R$ avec A négatif, B positif et R négatif.

Cas 13 : $A + B = R$ avec A, B, R négatifs.

Cas 14 et 15 : $A - B = R$ avec A, B négatifs, R positif ou négatif selon les valeurs de A et B.

Restent quatre cas anormaux :

Cas 1 : $A + B = R$ avec A, B positifs et R négatif.

Cas 7 : $A - B = R$ avec A positif, B négatif et R négatif.

Cas 10 : $A - B = R$ avec A négatif, B positif et R positif.

Cas 12 : $A + B = R$ avec A, B négatifs et R positif.

Ces quatre cas se produisent quand la valeur absolue du résultat dépasse la capacité de représentation dans la partie positive ou négative. En reprenant les valeurs de la figure 3.15, on peut illustrer les quatre cas anormaux (fig. 3.17).

Quand on réalise une opération d'addition ou de soustraction, si la fonction de débordement est à 0, le résultat est correct, en vraie grandeur. Si la fonction de débordement vaut 1, n eb ne suffisent pas pour représenter le résultat.

On peut déterminer la fonction débordement à partir du tableau de vérité de la figure 3.16.

$$DEB = \overline{OP}.(\bar{S}_A.\bar{S}_B.S_R + S_A.S_B.\bar{S}_R) + OP.\ (\bar{S}_A.S_B.S_R + S_A.\bar{S}_B.\bar{S}_R)$$

Cas (1)	+ 5	A	0	1	0	1	A			
	+ 6	B	+ 0	1	1	0	B			
	+ 11	R	1	0	1	1	R	Or 1011 correspond à − 5		
Cas (7)	+ 5	A	0	1	0	1	A			
	− 4	B	− 1	1	0	0	B			
	+ 9	R	1	0	0	1	R	Or 1001 correspond à − 7		
Cas (10)	− 5	A	1	0	1	1	A			
	+ 6	B	− 0	1	1	0	B			
	− 11	R	0	1	0	1	R	Or 0101 correspond à + 5		
Cas (12)	− 5	A	1	0	1	1	A			
	− 6	B	+ 1	0	1	0	B			
	− 11	R	0	1	0	1	R	Or 0101 correspond à + 5		

Fig. 3.17. — *Les cas de débordement.*

4.3. Traitement du débordement

Comme pour les nombres non signés, il peut y avoir un débordement anormal si les n eb suffisent par principe pour représenter toutes les valeurs possibles et un débordement normal si l'on traite des nombres à plus de n eb. La figure 3.18 montre une addition directe sur 24 eb avec une retenue sortante r_{24} égale à 1 et un débordement DEB_{24} égal à 0 (on applique la formule du débordement établie en 4.2, avec $OP = 0$, $S_A = 0$, $S_B = 1$, $S_R = 1$). Le débordement nul signifie que le résultat obtenu est correct en complément à 2.

	0 1 0 0	0 1 0 1	1 0 1 1	1 0 1 0	0 1 1 0	1 1 0 0	A
+	1 1 0 0	0 1 0 0	1 0 0 1	0 0 1 1	0 1 0 1	1 1 0 0	B
	0 0 0 0	1 0 1 0	0 1 0 0	1 1 0 1	1 1 0 0	1 0 0 0	S

$DEB_{24} = 0$ $R_{24} = 1$

Fig. 3.18. — *Addition directe de 2 nombres de 24 eb en complément à 2.*

Si l'on réalise la même addition avec un additionneur à 8 eb, pourvu d'une retenue sortante et d'un débordement on obtient l'enchaînement de la figure 3.19 :

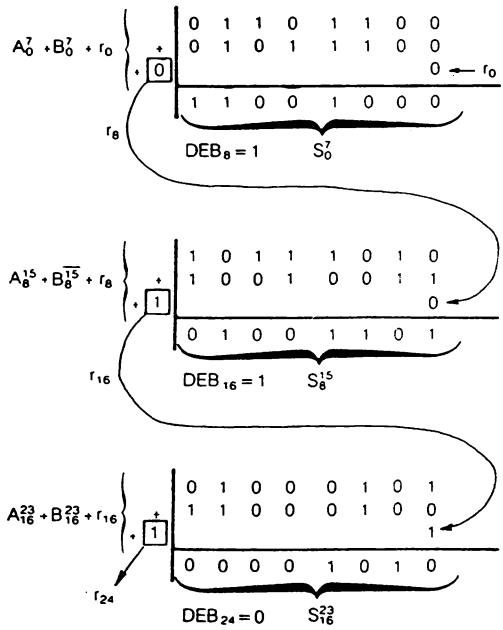


Fig. 3.19. — *Addition de 2 nombres de 24 eb en trois étapes.*

- Pour la première tranche, un débordement égal à 1 (on additionne deux nombres positifs et l'on obtient un résultat négatif) et une retenue nulle. On ne tient pas compte du débordement et l'on applique la retenue sortante en entrée de la deuxième tranche.
- Pour la deuxième tranche, un débordement égal à 1, dont on ne tient pas compte et une retenue sortante égale à 1, que l'on applique à la troisième tranche.
- Après addition de la troisième tranche, on obtient un débordement nul qui indique que le résultat est correct en complément à 2, et une retenue égale à 1, dont on ne tient pas compte, puisqu'il n'y a plus rien à ajouter.

5. ADDITION, SOUSTRACTION DE NOMBRES EN VIRGULE FIXE

On procède exactement comme pour des nombres en complément à 2 (fig. 3.20).

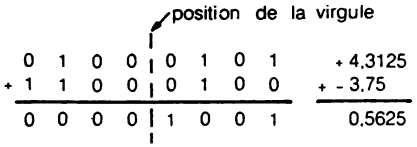


Fig. 3.20. — *Addition en virgule fixe.*

On trouve les mêmes conditions aux limites que pour les nombres entiers représentés en complément à 2.

6. ADDITION, SOUSTRACTION EN VIRGULE FLOTTANTE

Soit deux nombres: $A = M_A.2^{E_A}$, $B = M_B.2^{E_B}$, en virgule flottante. Cette forme se prête très bien à la multiplication ou à la division.

$$A.B = M_A.M_B.2^{E_A + E_B}$$

$$\frac{A}{B} = \frac{M_A}{M_B}.2^{E_A - E_B}$$

mais pas tellement bien à l'addition ou la soustraction. Étudions-en le principe.

Supposons que $E_A > E_B$ et que $E_A = E_B + n$. On peut écrire:

$$A = M_A.2^{E_A},\ B = M_B.2^{E_A - n} = M_B.2^{-n}.2^{E_A}$$

$$A + B = \left(M_A + \frac{M_B}{2^n}\right)2^{E_A}$$

Pour additionner deux nombres en virgule flottante, il faut:

— comparer les exposants et calculer leur différence n.

— diviser la mantisse du plus petit par 2^n (voir § 7.1).

— additionner la mantisse du premier et la nouvelle mantisse du second.

Ceci est assez compliqué et nous n'irons pas plus loin dans cette analyse. Sachons qu'il existe des blocs de calculs travaillant en flottant.

7. ALGORITHMES DE LA MULTIPLICATION

7.1. Multiplication et division par des puissances de 2 de nombres non signés

La multiplication ou la division par une puissance n de 2 revient à déplacer la virgule de n rangs vers la droite ou vers la gauche. L'opération est en fait un décalage.

$$\underbrace{0\,0\,0\,1\;\;0\,1\,0\,0}_{20} \times \underset{8}{2^3} \Rightarrow \underbrace{1\,0\,1\,0\;\;0\,0\,0\,0}_{160}$$

$$\underbrace{0\,0\,0\,1\;\;0\,1\,0\,0}_{20} : \underset{4}{2^2} \Rightarrow \underbrace{0\,0\,0\,0\;\;0\,1\,0\,1}_{5}$$

Comme les nombres traités ont une taille limitée, un décalage trop important fait perdre des eb significatifs et la multiplication et la division ne sont plus correctes.

$$\underbrace{0\,0\,0\,1\;\;0\,1\,0\,0}_{20} \times \underset{16}{2^4} \Rightarrow \underbrace{0\,1\,0\,0\;\;0\,0\,0\,0}_{64}$$

$$\underbrace{0\,0\,0\,1\;\;0\,1\,0\,0}_{20} : \underset{8}{2^3} \Rightarrow \underbrace{0\,0\,0\,0\;\;0\,0\,1\,0}_{2}$$

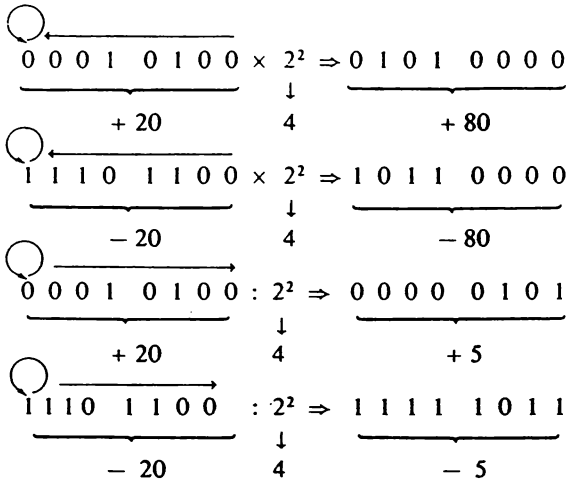
7.2. Multiplication et division par des puissances de 2 de nombres signés en complément à 2

La multiplication et division de nombres binaires représentés en complément à 2 suit les mêmes règles que celles des nombres non signés avec comme condition supplémentaire :

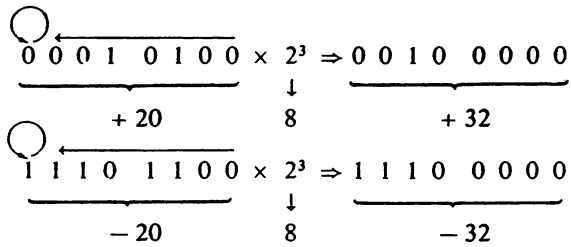
— De conserver l'eb de signe dans un décalage à gauche, de façon à ne pas changer le signe du nombre.

— De conserver et propager l'eb de signe dans un décalage à droite.

Exemples :



Si le décalage est trop important par rapport à la taille limitée des nombres, la multiplication n'est plus correcte.



7.3. Algorithme de la multiplication de deux nombres non signés

7.3.1. Algorithme manuel

Prenons deux nombres X et Y à n eb chacun. On montre que le produit XY a besoin de $2n$ eb pour être représenté. Réalisons la multiplication de X par Y selon l'algorithme classique utilisé au paragraphe 1.1.

$X = X_{n-1}\ X_{n-2} \ldots X_1\ X_0$

$Y = Y_{n-1}\ Y_{n-2} \ldots Y_1\ Y_0$

On réalise la somme des produits partiels suivants :

$P_0 = X.Y_0$

$P_1 = (X$ décalé à gauche de 1 pas). Y_1

Soit d'après le paragraphe 7.1 :

$P_1 = X.2^1.Y_1$

$P_2 = (X$ décalé à gauche de 2 pas). Y_2

Soit $P_2 = X.2^2.Y_2$

Etc.

$P_{n-1} = X.2^{n-1}.Y_{n-1}$

D'où finalement l'expression du produit :

$P = X.Y_0 + X.Y_1.2^1 + X.Y_2.2^2 + \ldots + X.Y_{n-1}.2^{n-1}$

La figure 3.21 qui applique cet algorithme montre qu'au fur et à mesure que l'on élabore des produits partiels, la taille des nombres à additionner augmente. Pour le dernier terme, il faut un additionneur de $2n-1$ eb.

Rang				3	2	1	0	
Nombre X				1	1	0	1	
Nombre Y				1	0	1	1	
				1	1	0	1	Produit partiel $P_0(XY_0)$
			+ 1	1	0	1		Produit partiel $P_1(XY_1.2^1)$
		1	0	0	1	1	1	Somme partielle
	+ 0	0	0	0				Produit partiel $P_2(XY_2.2^2)$
		1	0	0	1	1	1	Somme partielle
	+ 1	1	0	1				Produit partiel $P_3(XY_3.2^3)$
1	0	0	0	1	1	1	1	Produit final

Fig. 3.21. — *Algorithme classique de la multiplication.*

7.3.2. Algorithme par décalage des sommes partielles

Il est possible de réaliser une multiplication de n eb par n eb avec un additionneur à n eb seulement, en utilisant un algorithme différent. Plutôt

que de décaler le produit partiel $X.Y_i$ vers la gauche pour l'additionner à la somme partielle précédemment obtenue, on décale la somme partielle vers la droite et on lui additionne le produit partiel. La figure 3.22 illustre ce mécanisme.

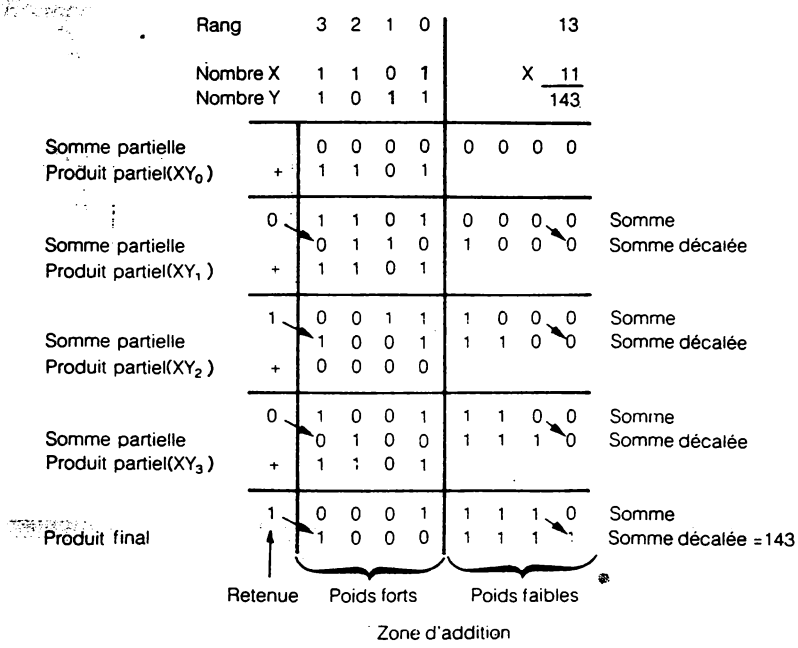


Fig. 3.22. — *Algorithme de multiplication avec décalage des résultats partiels.*

On pourrait démontrer que la somme finale a exactement la même forme que celle obtenue avec l'algorithme classique, c'est-à-dire :

$$P = X.Y_0 + X.Y_1.2^1 + X.Y_2.2^2 + \ldots + X.Y_{n-1}.2^{n-1}.$$

7.4. Algorithme de multiplication directe de nombres signés en complément à 2

Prenons deux nombres X et Y en complément à 2 et analysons les différents cas selon les signes de X et Y.

— *X et Y positifs*

Les eb de signe sont nuls, les nombres ont une forme pondérée iden-

tique à celle des nombres non signés, qui seraient précédés d'un 0. Les algorithmes développés en 7.3 conviennent.

— *X négatif et Y positif*

Le multiplicateur étant positif, Y a la forme pondérée normale d'un nombre non signé, précédé d'un 0. La multiplication revient donc à faire une suite de sommes de X multipliée par des puissances successives de 2. Ces sommes portent sur des nombres négatifs (puisque X est négatif), qui doivent être exprimés sur $2n$ eb. On doit donc faire une "extension du signe" comme le montre la figure 3.23. On obtient le même résultat en utilisant l'algorithme du décalage des sommes partielles vu au paragraphe 7.3.2 à condition de remplacer les décalages logiques incluant la retenue par des décalages arithmétiques (fig. 3.24).

Rang				3	2	1	0	- 3	
Nombre X				1	1	0	1	X + 5	
Nombre Y				0	1	0	1	- 15	
(1	1	1	1)	1	1	0	1	Produit partiel XY_0 avec signe étendu	
(0	0	0)	0	0	0	0	0	Produit partiel XY_1 avec signe étendu	
1	1	1	1	1	1	0	1	Somme partielle	
(1	1)	1	1	0	1			Produit partiel XY_2 avec signe étendu	
1	1	1	1	1	0	0	0	1	Somme partielle
(0)	0	0	0	0				Produit partiel XY_3 avec signe étendu	
(1)	1	1	1	0	0	0	1	Produit final	
↑									
Signe				Valeur					

Les chiffres entourés correspondent à l'extension du signe

Fig. 3.23. — *Multiplication d'un nombre X négatif par un nombre Y.*

— *X négatif ou positif, Y négatif*

La multiplication d'un nombre X positif ou négatif par un nombre Y positif revient à une suite d'additions de X décalés. Ceci est dû au fait qu'en complément à 2, un nombre positif a exactement la même forme pondérée qu'un nombre non signé. Il est simplement précédé d'un eb à 0.

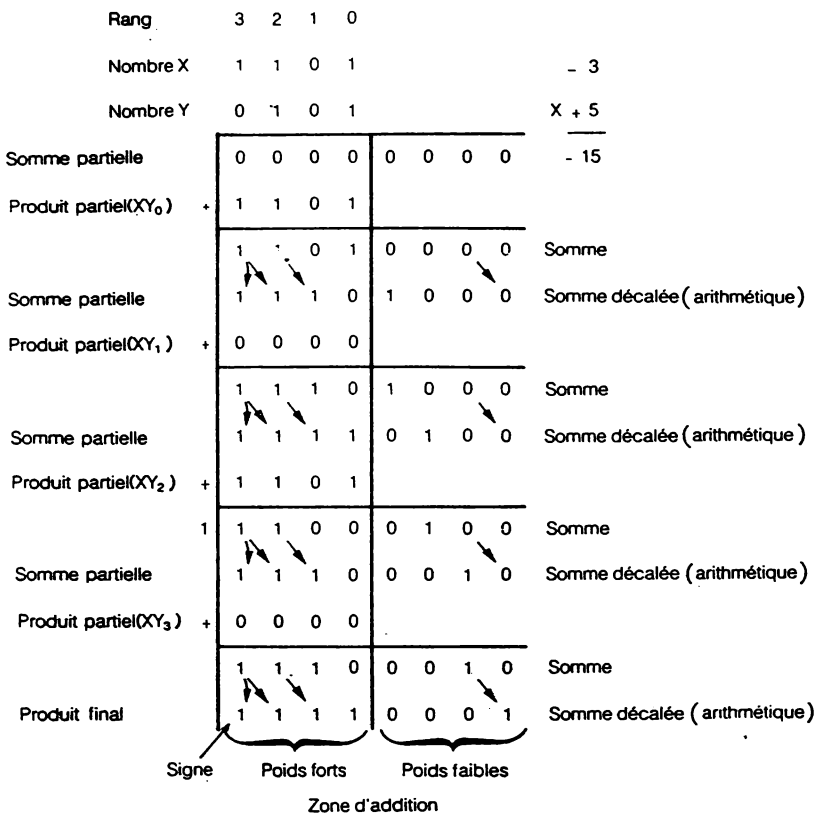


Fig. 3.24. — *Multiplication d'un nombre X négatif par un nombre Y positif avec l'algorithme de décalage des sommes partielles.*

A priori, un nombre négatif en complément à 2 perd cette forme pondérée et il ne semble pas possible d'effectuer la multiplication selon les mêmes algorithmes. En fait, c'est possible, avec une légère variante. On peut montrer qu'un nombre négatif a aussi une forme pondérée, en affectant des poids positifs à tous les eb de valeur, et un poids négatif à l'eb de signe. Si l'on se reporte à la figure 3.15 et si l'on donne la valeur -2^3 à l'eb de signe, on vérifie par exemple que :

$$-3 = -2^3 + 1.2^2 + 0.2^1 + 1.2^0 = -8 + 4 + 1 = -3.$$

Cette remarque permet d'écrire un nombre en complément à 2 sous une forme pondérée qu'il soit positif ou négatif.

$$Y = Y_{n-1}\ Y_{n-2} \ldots Y_1\ Y_0$$

équivaut à :

$$Y = -\ Y_{n-1}.2^{n-1} + Y_{n-2}.2^{n-2} + \ldots + Y_1.2^1 + Y_0.2^0.$$

Cette forme convient aussi bien pour un nombre positif dans lequel $Y_{n-1} = 0$, ce qui élimine le poids 2^{n-1} que pous un nombre négatif dans lequel $Y_{n-1} = 1$, ce qui donne bien le poids négatif -2^{n-1} à l'eb de signe.

La multiplication de X par Y s'écrit alors :

$$X.Y = X.(-\ Y_{n-1}.2^{n-1} + Y_{n-2}.2^{n-2} + \ldots + Y_1.2^1 + Y_0.2^0)$$

$$= \underbrace{-\ X.Y_{n-1}.2^{n-1}}_{\text{Décalage de } X \text{ et soustraction}} + \underbrace{X.(Y_{n-2}.2^{n-2} + Y_{n-3}.2^{n-3} + \ldots + Y_1.2^1 + Y_0.2^0)}_{\text{Suite d'additions et décalages de } X}$$

Compte tenu de la forme du produit $X.Y$, on voit que l'on peut appliquer les algorithmes développés au paragraphe 7.3 en remplaçant la dernière opération par une soustraction (si $Y_{n-1} = 1$). La figure 3.25 montre un exemple de produit de deux nombres négatifs.

Rang		3	2	1	0					
Nombre X		1	1	0	1					- 3
Nombre Y		1	1	0	1					X - 3
Somme partielle		0	0	0	0	0	0	0	0	+ 9
Produit partiel XY_0	+	1	1	0	1					
		1	1	0	1	0	0	0	0	Somme
Somme partielle		1	1	1	0	1	0	0	0	Somme décalée (arithmétique)
Produit partiel XY_1	+	0	0	0	0					
		1	1	1	0	1	0	0	0	Somme
Somme partielle		1	1	1	1	0	1	0	0	Somme décalée (arithmétique)
Produit partiel XY_2	+	1	1	0	1					
	0	1	1	0	0	0	1	0	0	Somme
Somme partielle		1	1	1	0	0	0	1	0	
Produit partiel XY_3	-	1	1	0	1					
		0	0	0	1	0	0	1	0	Différence
Produit final		0	0	0	0	1	0	0	1	

Fig. 3.25. — *Multiplication de deux nombres négatifs avec l'algorithme de décalage des sommes partielles.*

En conclusion, on peut réaliser la multiplication de nombres signés en complément à 2 avec les algorithmes manuels ou de décalages de sommes, à condition de remplacer la dernière somme par une soustraction si le multiplicateur est négatif.

7.5. Amélioration des performances de la multiplication. Algorithme de Booth

7.5.1. Principe

L'algorithme de Booth pour la multiplication des nombres signés en complément à 2 améliore les performances en réduisant le nombre d'additions à effectuer. Il analyse les eb du multiplicateur et tient compte de l'existence de suites consécutives de 0 ou de 1.

- S'il y a une suite de zéros, il n'y a pas à additionner mais seulement à décaler (procédure déjà utilisée dans l'algorithme manuel du paragraphe 1.1).
- S'il y a une suite de uns, dont le premier correspond au rang m et le dernier au rang p, on peut remplacer la suite d'additions et de décalages, effectuées dans l'algorithme classique.

$$S = X2^p + X2^{p-1} + \ldots + X2^{m+1} + X2^m = X.(2^p + 2^{p-1} + \ldots + 2^{m+1} + 2^m)$$

par les mêmes décalages et une addition et une soustraction seulement.

$$S = X.(2^{p+1} - 2^m)$$

En effet :

			$p+1$ ↓	p ↓					m ↓		
	0	0	1	0	0	0		0	0	0	0
−	0	0	0	0	0	0	...	0	1	0	0
	0	0	0	1	1	1	...	1	1	0	0

Chaîne de 1 du rang m au rang p

Pour mettre en œuvre cet algorithme, on analyse donc la succession des eb du multiplicateur. La figure 3.26 donne les opérations à faire pour les quatre cas possibles de deux eb successifs.

eb du multiplicateur		Opération	Décalage
Y_i	Y_{i-1}		
0	0	-	Oui
1	0	Soustraction	Oui
1	1	-	Oui
0	1	Addition	Oui

Fig. 3.26. — *Analyse des eb du multiplicateur et actions dans l'algorithme de Booth.*

7.5.2. Étude des différents cas

Examinons les différents cas d'application de l'algorithme de Booth.

— *X positif ou négatif, Y positif*

Y a la forme pondérée habituelle, la multiplication est une suite d'additions et de soustractions de *X*, représenté en complément à 2, décalé à gauche. On obtient le résultat en complément à 2. La figure 3.27 donne un exemple d'application. Un eb fictif Y_{-1} est introduit pour débuter le calcul.

										Eb considérés	Opération	Décalage X
- 5	Rang	3	2	1	0							
X + 6	Nombre X	1	0	1	1				Y_{-1}			
- 30	Nombre Y	0	1	1	0	0						
Somme partielle (initiale)		0	0	0	0	0	0	0	0			
		.	.	.	.	.	.	.	.	$Y_0Y_{-1} = 00$	Aucune	Oui
Somme partielle (inchangée)		0	0	0	0	0	0	0	0			
Signe étendu →	-	1	1	1	1	0	1	1		$Y_1Y_0 = 10$	Soustraction	Oui
Somme partielle		0	0	0	0	1	0	1	0			
		.	.	.	.	.	.			$Y_2Y_1 = 11$	Aucune	Oui
Somme partielle (inchangée)		0	0	0	0	1	0	1	0			
	+	1	1	0	1	1				$Y_3Y_2 = 01$	Addition	Non
Résultat final (-30)		1	1	1	0	0	0	1	0			

Fig. 3.27. — *Multiplication d'un nombre négatif par un nombre positif selon l'algorithme de Booth.*

— *X positif ou négatif, Y négatif*

Y n'a plus la forme pondérée habituelle puisque l'eb de signe a un poids négatif. On distingue deux cas : l'eb de signe qui vaut 1 est précédé à droite d'un 0 ou d'une suite de 0, ou l'eb de signe est précédé par un 1 ou une suite

de 1 à droite. Si l'eb de signe est précédé à droite d'un 0, ou d'une suite de 0, on constate à la figure 3.28 que pour :

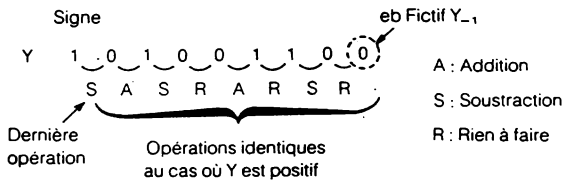


Fig. 3.28.

tous les eb précédant à droite l'eb de signe, les opérations sont identiques à ce que l'on aurait fait pour un nombre Y positif. La dernière opération réalisée avec l'eb de signe est une soustraction, ce qui correspond bien à affecter un poids négatif à l'eb de signe.

Si l'eb de signe est précédé d'un 1 ou d'une suite de 1, à la figure 3.29, on constate que les opérations sont identiques jusqu'au premier 1 de la chaîne de 1 :

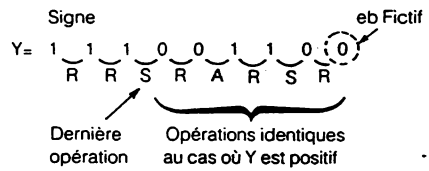


Fig. 3.29.

qui précède à droite l'eb de signe. La dernière opération est une soustraction. Comment a-t-on tenu compte du poids négatif de l'eb de signe ? Ce n'est pas très évident. Supposons que le premier 1 de la chaîne ait le rang p et que l'eb de signe ait le rang $n-1$ (fig. 3.30). Avec l'algorithme de multi-

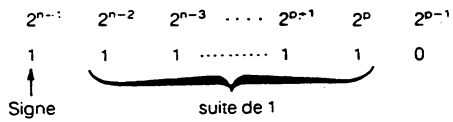


Fig. 3.30.

plication des nombres négatifs du paragraphe 7.4., il faut effectuer normalement l'opération :

$$S = -X.2^{n-1} + X.2^{n-2} + X.2^{n-3} + \ldots + X.2^{p+1} + X.2^{p}$$

Cette série vaut $-X.2^{p}$ comme le montre le calcul ci-dessous :

$$S = -X.2^{n-1} + X.(2^{n-2} + 2^{n-3} + \ldots + 2^{p+1} + 2^{p})$$

$$S = -X.2^{n-1} + X.(2^{n-1} - 2^{p})$$

$$S = -X.2^{p}$$

Or, en appliquant l'algorithme de Booth, quand on rencontre le premier 1 d'une chaîne, au rang p, on fait une soustraction de $X2^{p}$, ce qui correspond bien au résultat souhaité. La figure 3.31 montre en application la multiplication de deux nombres négatifs.

		3	2	1	0						
- 5											
X - 3	Nombre X	1	0	1	1		Y_{-1}			Eb considérés	Opération	Décalage X
+ 15	Nombre Y	1	1	0	1	(0)					
Somme partielle (initiale)		0	0	0	0	0	0	0	0			
	-	(1	1	1	1)	1	0	1	1	$Y_0Y_{-1} = 10$	Soustraction	Oui
Somme partielle		0	0	0	0	0	1	0	1			
	+	(1	1	1	1)	0	1	1		$Y_1Y_0 = 01$	Addition	Oui
Somme partielle		1	1	1	1	1	0	1	1			
	-	(1 | 1)	1	0	1	1			$Y_2Y_1 = 10$	Soustraction	Oui
Résultat final (-15)		0	0	0	0	1	1	1	1	$Y_3Y_2 = 11$	Aucune	Non

Fig. 3.31. — *Multiplication de deux nombres négatifs selon l'algorithme de Booth.*

L'intérêt de l'algorithme de Booth est d'autant plus grand que le multiplicateur comprend de longues suites de 1 ou de 0, puisque l'on supprime des opérations. Si les 1 et 0 sont alternés fréquemment, le gain est plus faible ou même nul.

7.6. Multiplication ramenée à des nombres ± |N|

Une autre façon de multiplier des nombres signés en complément à 2 est de les ramener à des nombres exprimés sous la forme signe et valeur absolue. On calcule la valeur absolue du résultat et le signe du résultat sépa-

rémment. On repasse ensuite le résultat en complément à 2. Les étapes sont les suivantes :

a) Calculer le signe du résultat :

Soit S_A et S_B les signes A et B, le tableau de vérité de la figure 3.32 donne le signe $S_R = S_A.\overline{S}_B + \overline{S}_A.S_B$.

S_A	S_B	S_R
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 3.32. — *Tableau de vérité du signe en multiplication.*

b) Si les deux nombres sont positifs ($S_A + S_B = 0$), on prend les $n - 1$ eb représentant la valeur de chacun des deux nombres et l'on effectue la multiplication suivant le principe exposé au paragraphe 7.3. c'est-à-dire par une série de décalages et d'additions. On juxtapose ensuite l'eb de signe 0 au résultat.

c) Si un nombre ou les deux sont négatifs ($S_A + S_B = 1$), on complémente à 2 les nombres négatifs et l'on effectue le même calcul que précédemment, les nombres étant alors positifs. Si le signe du résultat est négatif, on complémente à 2 les eb de valeur.

La figure 3.33 montre ces opérations pour deux cas.

La multiplication effectuée suivant cette méthode est complexe à mettre en œuvre et nécessite une succession d'opérations élémentaires : calcul du signe, complémentation des nombres négatifs, multiplication, complémentation éventuelle du résultat, compléter le nombre résultant avec son signe. Plus facile à comprendre sur le plan des principes, cette méthode est plus lente pour les performances. On l'utilise surtout dans des traitements programmés.

CALCULS \ NOMBRES	(+9) X (+11) (01001) X (01011)	(-9) X (+11) (10111) X (01011)
Signe résultant	$S_A\overline{S_B} + \overline{S_A}S_B = 0$	$S_A\overline{S_B} + \overline{S_A}S_B = 1$
Les nombres sont-ils positifs ?	$S_A + S_B = 0$ (Oui)	$S_A + S_B = 1$ (Non)
Complémentation des nombres négatifs		(01001) X (01011)
Multiplication sur des nombres positifs (résultat sur 8 eb) 3 décalages et 4 additions partielles	1 0 0 1 1 0 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 1 (99)	
Complémenter le nombre si le signe résultant est négatif (8 eb)		10011101
Compléter avec le signe (résultat sur 9 eb)	001100011	110011101

Fig. 3.33. — *Multiplication ramenée à des nombres* $\pm$ $|N|$.

Chapitre 4

RÉALISATION ÉLECTRIQUE DES FONCTIONS LOGIQUES

1. PRÉSENTATION DES FONCTIONS LOGIQUES EN BOITIER

La figure 4.1 montre deux schémas de portes logiques NAND réalisées avec des résistances, diodes et transistors. Le premier représente un montage appelé DTL (Diode Transistor Logic), car son réseau d'entrée comprend des diodes et son réseau de sortie un transistor. Le second représente un montage TTL (Transistor Transistor Logic) car le réseau d'entrée comprend un transistor (à émetteur multiple) comme le réseau de sortie.

Construites avec des éléments discrets (transistors, diodes, résistances), des portes comme celles-ci occupaient une surface de quelques dizaines de centimètres carrés sur une carte support. Vers 1965, les constructeurs de composants commencèrent à mettre sur un même substrat de silicium tous les composants nécessaires : diodes, transistors, résistances, et réalisèrent ainsi les premiers circuits logiques "intégrés". Les fonctions logiques furent alors proposées groupées dans un "boîtier" possédant de nombreuses broches pour accéder aux entrées et aux sorties des opérateurs logiques. La figure 4.2 donne l'aspect de quelques boîtiers. Les boîtiers les plus courants comportent 14, 16, 18, 20, 24, 28, 40 broches disposées en deux rangées parallèles. Il existe aussi d'autres formes que celles de la figure 4.2 : boîtiers

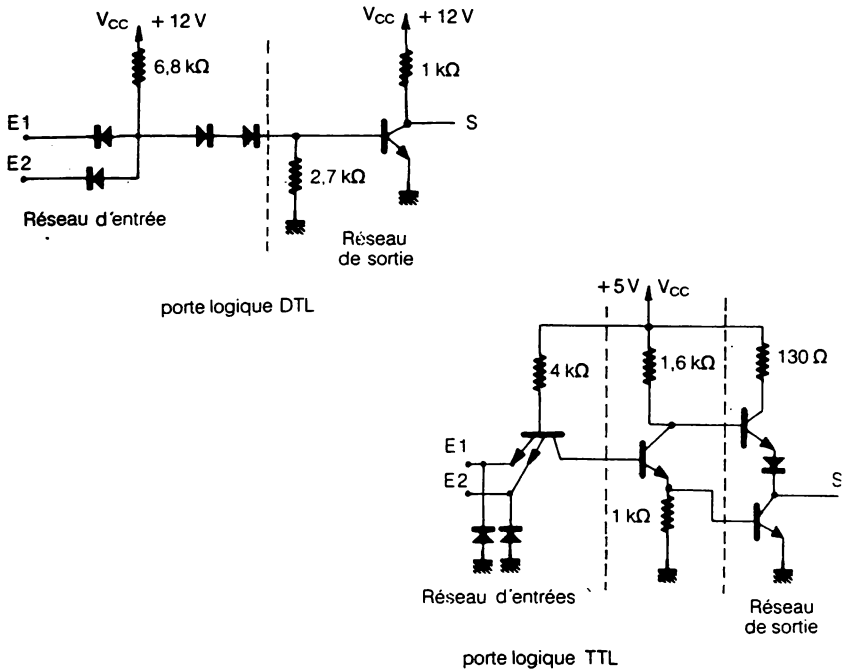


Fig. 4.1. — *Schéma de portes logiques NAND.*

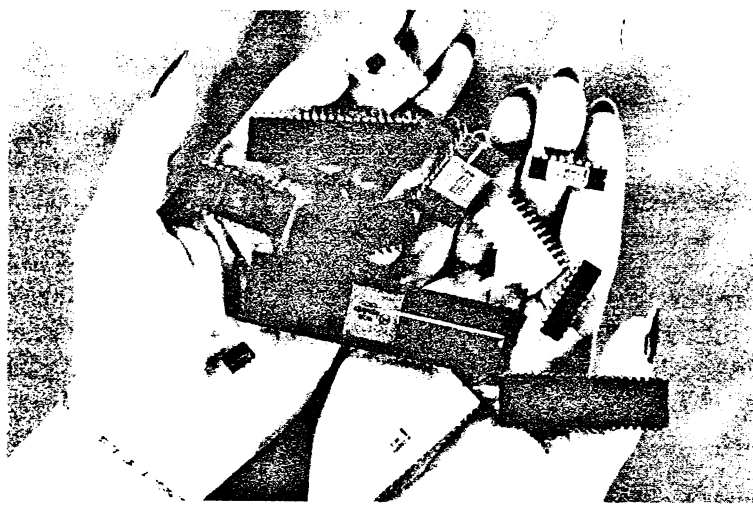


Fig. 4.2. — *Aspects des boîtiers intégrés.*

plats de dimensions plus réduites, boîtiers ayant quatre rangées de broches sur les quatre côtés de la forme rectangulaire, boîtiers ayant deux double rangées sur les deux grands côtés du rectangle, boîtiers porteurs sans broches, etc. La partie active du boîtier (transistor, dicdes, etc...) est moulée dans un enrobage en plastique ou en céramique.

Un boîtier de 14 broches mesure environ 20 mm de long sur 8 mm de large, et le moins complexe contient quatre portes logiques. Par rapport à la surface d'une porte à éléments discrets, c'est au moins un facteur 100 qui a été gagné.

La figure 4.3 montre comment sont distribuées les 14 broches d'un boîtier contenant quatre portes NAND à deux entrées vu de dessus:

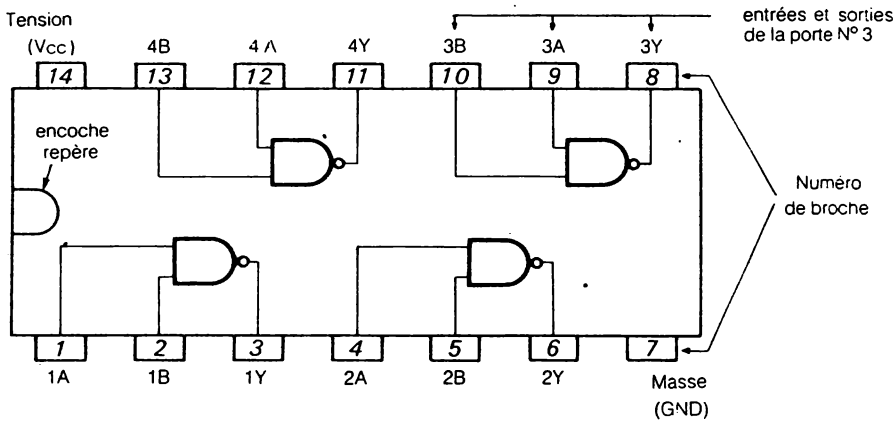


Fig. 4.3. — *Exemple de brochage d'un boîtier.*

Chaque boîtier logique est désigné par un symbole alphanumérique qui donne l'identité du composant. Les quatre portes NAND à deux entrées contenues dans un boîtier sont désignées par:

SN 74 *LS* 00 *N*

SN indique qu'il s'agit d'une série standard (par opposition à des familles à caractéristiques spéciales: résistance aux radiations spécifications particulières de fiabilité, ...).

74 indique qu'il s'agit de la série professionnelle, travaillant dans la gamme de température de 0° C à 70° C.

LS signifie "Low Power Schottky" (famille de circuits intégrés).

00 donne le numéro du boîtier dans la famille.

N indique le type de boîtier dans lequel les fonctions sont encapsulées (*N*: boîtier dit "dual-in line" avec deux rangées de broches et moulé en plastique).

Les boîtiers d'usage très courant sont la plupart du temps répertoriés de la même façon par des constructeurs différents. Les boîtiers plus spécifiques possèdent des identifications propres aux constructeurs.

Les boîtiers sont aussi classés selon leur densité d'intégration ou complexité interne. Quatre classes aux frontières assez floues sont couramment employées:

SSI : "Small Scale Integration": Intégration à faible densité. Cela correspond à quelques portes dans un boîtier.

MSI : "Medium Scale Integration": Intégration à densité moyenne. Cela correspond à un circuit comprenant l'équivalent de 12 à 100 portes logiques.

LSI : "Large Scale Integration": Intégration à haute densité. Cela correspond à un circuit comprenant l'équivalent de 100 à 1 000 portes dans un boîtier.

VLSI: "Very Large Scale Integration": Intégration à très haute densité. Au-delà de 1 000 portes. Les circuits les plus complexes conçus au début des années 1980 correspondent à 75 000 transistors ou dispositifs similaires.

2. LES FAMILLES DE CIRCUITS INTÉGRÉS

Un circuit intégré est rarement prévu pour fonctionner seul. La plupart du temps, on devra le relier à d'autres pour constituer un système. Pour que cette interconnexion soit la plus simple possible, les circuits appartiennent à une même famille logique qui donne à chacun de ses membres, des caractéristiques communes. Quand on doit, dans un même système utiliser deux familles de circuits, il faut prévoir des interfaces de passage dans les deux sens.

2.1. Définition d'une famille logique

Comment définir une famille logique? Il y a de multiples critères... Citons:

- Le procédé de fabrication: utilisation de silicium, de matériau non cristallin, silicium sur saphir, etc...
- Le phénomène utilisé: déplacement de porteurs dans un milieu semi-conducteur (exploitation de l'effet transistor), de charges électriques dans une structure, de domaines magnétiques, etc...
- Le schéma électrique définissant la porte élémentaire. Divers schémas ont été proposés: RTL (Résistance Transistor Logic), porte constituée de résistances et d'un transistor; DTL (Diode Transistor Logic), porte avec diodes en entrée, résistance et un transistor; TTL (Transistor Transistor Logic) où le circuit d'entrée à diodes est remplacé par un transistor à émetteurs multiples; ECL (Emitter Coupled Logic) où les circuits travaillent en régime linéaire et non plus saturé-bloqué, IIL ou I^2L (logique à injection de courant).
- La puissance consommée par la porte standard de la famille ou par une structure répétitive dans un circuit (consommation d'une cellule mémoire).
- La vitesse de fonctionnement de la porte standard caractérisée par le temps moyen de transit d'un signal au travers de la porte. Ce paramètre dépend fortement du précédent. A technologie donnée, plus une porte est rapide, plus elle consomme d'énergie.
- La température de fonctionnement. Il existe deux gammes: professionnelle de 0 à 70° C et militaire de $-55°$ C à $+125°$ C.

La pratique a fait que les familles logiques ont été caractérisées par l'un ou l'autre des attributs précédents.

2.2. Les familles logiques couramment utilisées

2.2.1. Famille TTL

C'est la famille la plus développée. Elle date de la fin des années 1960. Elle a connu de nombreuses évolution depuis, dans le domaine de la vitesse.

de la consommation, et de la variété des fonctions proposées. Elle constitue un standard même pour les autres familles qui définissent souvent leurs produits avec des interfaces compatibles TTL.

2.2.2. Famille ECL

Réservée aux traitements ultra-rapides (portes avec des retards voisins ou inférieurs à la nano seconde (10^{-9} s), délicate d'emploi à cause des problèmes d'interconnexion. Elle est maintenant souvent employée au cœur des circuits très complexes, apportant ainsi un gain notable dans les performances, la périphérie du circuit étant compatible TTL, ce qui simplifie les liaisons.

2.2.3. Famille MOS

(Metal Oxide Semi-conductor). Alors que les deux familles précédentes TTL et ECL utilisent des transistors bipolaires, celle-ci fonctionne avec des transistors unipolaires à effet de champ. Le gros intérêt de cette famille réside en ce que les éléments actifs sont beaucoup plus petits que les éléments actifs bipolaires et qu'ils consomment moins. Cela permet de réaliser des circuits LSI ou VLSI comme des mémoires ou des microprocesseurs. Par contre, il n'y a pas de portes logiques ou de circuits MSI en technologie MOS car ils seraient trop lents.

2.2.4. Famille CMOS

(Complementary Metal Oxide Semi conductor). En associant des transistors MOS à canal *P* et à canal *N*, on peut réaliser des portes logiques qui ne consomment de manière significative que lors des commutations pour charger ou décharger les capacités parasites et très peu au repos. Il y a quelques familles CMOS dont certaines reprennent les circuits de la famille TTL. Cela a l'avantage de restreindre la mise au courant pour un concepteur déjà utilisateur de TTL. Cette famille s'utilise lorsque la consommation d'énergie doit être très basse. Elle possède par ailleurs une très bonne immunité au bruit : les parasites peuvent atteindre 45 % de la tension d'alimentation. Les familles CMOS les plus récentes ont la vitesse de la TTL-LS.

3. LES FAMILLES TTL (TRANSISTOR-TRANSISTOR LOGIC)

Nous nous intéressons maintenant à la famille la plus utilisée. Conçue à la fin des années 1960 par Texas Instruments, la famille TTL a beaucoup évolué et donné naissance à plusieurs sous-familles.

La première est la série TTL dite standard ou normale. Le schéma de la porte de base, la porte NAND correspond au schéma de la figure 1. Le nom TTL vient de ce que dans ce schéma, les réseaux d'entrée et de sortie sont constitués de transistors, alors que dans la génération précédente dite DTL, le réseau d'entrée était à diodes et le réseau de sortie à transistor. La porte TTL standard commute (c'est-à-dire, passe de l'état 0 à l'état 1 ou inversement, sous l'action d'un changement d'état d'une de ses entrées) en moyenne en 10 nS et consomme 10 mW.

La seconde sous-famille fut la TTL série H (High Speed), plus rapide que la précédente (commutation en 6 nS) mais aussi plus gourmande en énergie (22 mW par porte).

Une troisième sous-famille fut introduite: TTL série L (Low Power) pour les applications où diminuer l'énergie consommée est primordiale. La porte de base ne consomme que 1 mW mais réagit plus lentement, en 33 nS.

Ensuite, Texas Instruments améliora très nettement les performances de commutation des portes par l'adjonction de diodes "Schottky" aux structures TTL existantes. Ces diodes placées entre la base et le collecteur de transistor empêchent leur sursaturation et augmentent la vitesse de commutation. La TTL série S (comme Schottky) consomme 19 mW par porte (à peu près autant que la série H) mais commute deux fois plus vite, en 3 nS.

Enfin, le même constructeur proposa la série TTL-LS (LS comme "Low Power Schottky"), où en conservant la même vitesse de commutation que la série standard (10 nS pour une porte), les diodes Schottky et d'autres raffinements faisaient chuter la consommation à 2 mW par porte.

On notera que si les premiers circuits de série méritent le nom de TTL-LS à cause du réseau d'entrée à émetteur multiples, d'autres circuits de la série LS utilisent des réseaux d'entrées différents, notamment à diodes, comme les anciennes portes DTL.

La gamme des circuits de la série LS fut d'abord obtenue en reprenant avec la conception LS les circuits SSI et MSI les plus utilisés de la série standard. Le concepteur retrouvait ainsi des circuits bien connus mais

consommant 5 fois moins à vitesse égale. Puis, les constructeurs introduisirent de nouveaux circuits LSI car l'élément de base consommant moins, sa taille peut être plus petite, et à puissance consommée constante, on peut intégrer plus de circuits sur le même support.

A la fin de l'année 1979 et au début 1980, les constructeurs ont introduit de nouvelles familles succédant à la LS, famille consommant moins (50 % environ) et allant plus vite (40 % plus rapide). Texas Instruments par exemple, commercialise ces circuits sous l'appellation ALS (Advanced Low Power) ou AS (Advanced Schottky) plus rapide. Signalons aussi la famille FAST, comparable à la famille AS.

4. FAMILLES LOGIQUES UTILISÉES

Dans les familles TTL, les versions standard H et L ne sont plus utilisées. La version TTL-LS a été très utilisée mais est en cours de remplacement par la version ALS.

Pour les familles rapides, on emploie les séries ECL (10 KH, 100 KH), les sous-familles TTL Schottky, S en voie de remplacement par AS, et FAST.

Enfin pour les faibles consommations les familles CMOS: HCMOS non-compatible TTL et HCT-MOS compatible TTL avec les performances dynamiques de la TTL-LS.

5. FONCTIONNEMENT D'UNE PORTE TTL STANDARD

5.1. Schéma

La figure 4.4 représente le schéma d'une porte TTL standard 7400. Les niveaux de sortie sont approximativement de 0 V et de + 5 V. Ce sont ces niveaux que nous supposerons appliqués aux entrées pour étudier le fonctionnement du circuit. Nous verrons plus loin comment doivent être corrigés ces niveaux dans la pratique. (Voir chapitre 5 : Paramètres électriques et temporels des circuits logiques.)

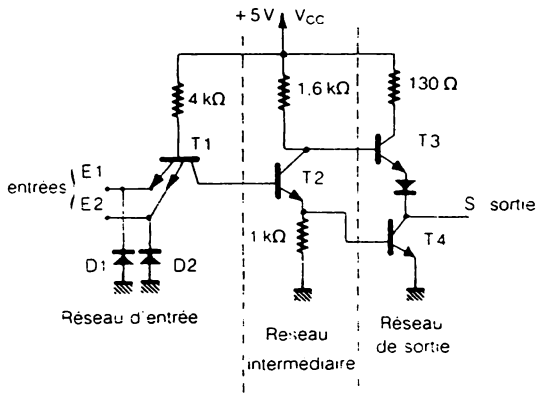


Fig. 4.4. – *Schéma électrique d'une porte NAND 7400.*

Le réseau d'entrée comprend un transistor à émetteur multiples. Les diodes D_1 et D_2 de la figure 4.4 ne jouent aucun rôle en temps normal, quand les signaux appliqués aux entrées sont positifs. Par contre, si ces signaux deviennent négatifs elles empêchent par écrétage, que le potentiel tombe trop bas. Ce sont des diodes de "clamping".

Le réseau de sortie, constitué de deux transistors et d'une diode "empilés" les uns au-dessus des autres a une structure dite "totem-pôle". On dit aussi qu'il s'agit d'une structure à charge active. La charge de T_4 est constituée par T_3 et réciproquement T_4 sert de charge à T_3.

5.2. Fonctionnement

a) Cas où E_1 ou E_2 ou E_1 et E_2 sont à la masse (fig. 4.5)

Quand E_1 ou E_2 ou les deux sont à la masse, le courant de base de T_1 passe entièrement dans les émetteurs et la base se trouve polarisée à un faible niveau qui ne permet pas à la jonction base collecteur de conduire. Il n'y a donc pas de courant base pour T_2 qui est bloqué. T_2 bloqué donne un potentiel 0 sur son émetteur, qui donne un courant de base nul pour T_4 qui est bloqué également et constitue pour T_3 une très forte impédance de charge d'émetteur. T_2 étant bloqué, la résistance de 1,6 k fournit un courant de base pour T_3 qui conduit. Le transistor T_3 ayant une forte impédance dans l'émetteur et une faible impédance dans son collecteur a un potentiel d'émetteur presque égal à la tension d'alimentation V_{CC}. La tension de sortie est donc voisine de V_{CC} quand E_1 ou E_2 ou E_1 et E_2 sont à 0 V.

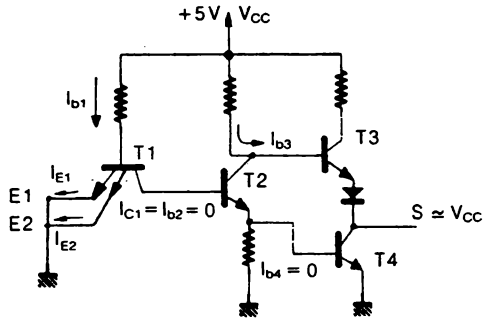


Fig. 4.5. — *Cas où les entrées sont à la masse.*

b) Cas où E_1 et E_2 sont à V_{CC} (fig. 4.6)

Quand E_1 et E_2 sont au + 5 V, les jonctions base-émetteurs sont bloquées et aucun courant ne les traverse, si ce n'est un faible courant inverse. Par contre, la jonction base collecteur est polarisée en direct et le courant de base I_{b1} se retrouve sur le collecteur de T_1, devient le courant de base de T_2 qui conduit. Le potentiel d'émetteur de T_2 polarise le transistor T_4 qui se sature et donne un potentiel voisin de la masse sur son collecteur. Le transistor T_3 sert de charge active au transistor T_4. La tension de sortie est donc voisine de 0 V quand les deux entrées sont à V_{CC} = + 5 V.

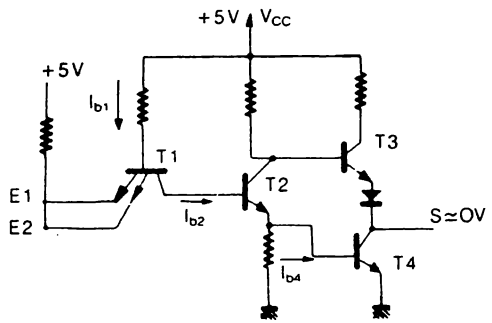


Fig. 4.6. — *Cas où les entrées sont au + 5 V.*

5.3. Table de vérité. Convention logique

On dresse une table de vérité entre les potentiels d'entrées et de sortie. Pour obtenir la fonction logique réalisée par la porte, on établit la correspondance entre les états électriques et les états logiques.

Il y a deux possibilités :

$0\ V \rightarrow 0$ logique
$+5\ V \rightarrow 1$ logique
} Convention de logique positive

$0\ V \rightarrow 1$ logique
$+5\ V \rightarrow 0$ logique
} Convention de logique négative

La convention de logique est positive si le 1 logique a le potentiel le plus haut, et négative dans le cas contraire. Dressons à la figure 4.7 les tableaux de vérité correspondants et exprimons les fonctions booléennes résultantes.

E1	E2	S
OV	OV	+5 V
OV	+5 V	+5 V
+5 V	OV	+5 V
+5 V	+5 V	OV

Table de vérité électrique

E1	E2	S
0	0	1
0	1	1
1	0	1
1	1	0

Table de vérité en logique positive

E1	E2	S
1	1	0
1	0	0
0	1	0
0	0	1

Table de vérité en logique négative

Fig. 4.7. — *Tableau de vérité d'une porte logique.*

En logique positive, nous obtenons :

$$S = \bar{E}_1.\bar{E}_2 + \bar{E}_1.E_2 + E_1.\bar{E}_2 = \bar{E}_1.(\bar{E}_2 + E_2) + \bar{E}_2.(\bar{E}_1 + E_1)$$

$$S = \bar{E}_1 + \bar{E}_2 = \overline{E_1.E_2}$$

C'est la fonction ET.NON ou NAND.

En logique négative, il vient :

$$S = \bar{E}_1.\bar{E}_2 = \overline{E_1 + E_2}$$

C'est la fonction OU.NON ou NOR.

6. PORTE A SORTIE COLLECTEUR OUVERT

Le schéma de la figure 4.4 avec entrée multi-émetteur et sortie de type "totem-pôle" est le schéma de base pour les circuits T.T.L. La figure 4.8

montre une variante dans laquelle on conserve le même schéma d'entrée mais où la sortie est réduite à un transistor sans charge dans son collecteur. Trois applications courantes illustrent l'utilisation de ce circuit.

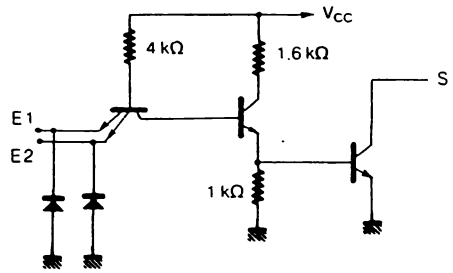


Fig. 4.8. — *Porte NAND à collecteur ouvert.*

6.1. Commande d'une charge externe

La porte à collecteur ouvert commande, à partir de signaux logiques (0 *V*, + 5 *V*) une charge indépendante du circuit alimentée soit sous la tension + 5 *V* comme le circuit lui-même (*exemple*: diode électroluminescente), soit sous une tension beaucoup plus élevée (30 *V*), avec un courant important (40 mA) dans la charge (fig. 4.9).

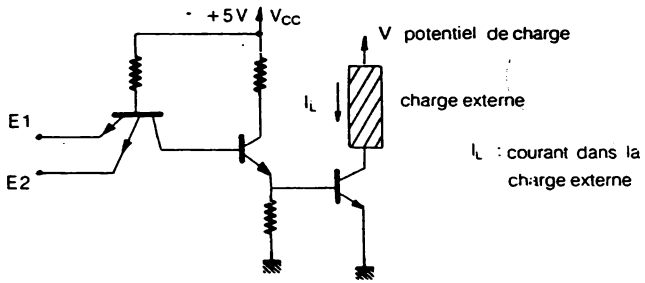


Fig. 4.9. — *Utilisation d'une porte NAND à collecteur ouvert pour alimenter une charge.*

6.2. Fonction ET câblé

Une seconde utilisation, héritée de la logique DTL, est le fonctionnement en ET câblé. Elle consiste à réaliser la fonction ET par simple

connexion entre les sorties, et mise en commun d'une résistance de charge (fig. 4.10). Examinons l'état du point S commun, en fonction de l'état des transistors a et b.

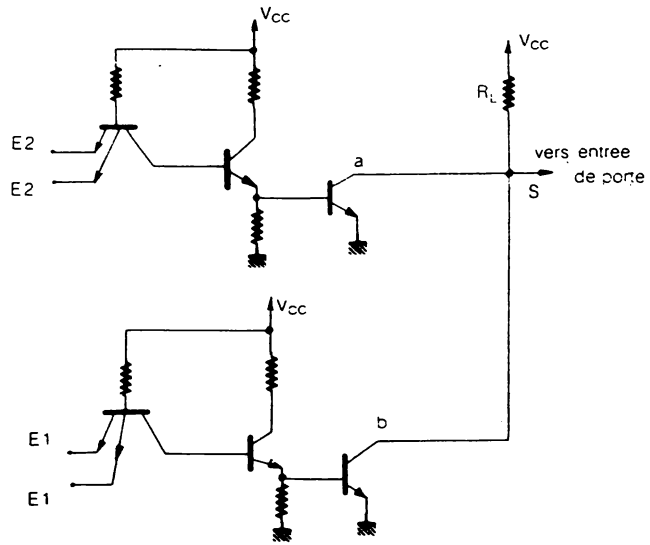


Fig. 4.10. — *Fonction ET câblée.*

1. a et b bloqués : Aucun courant ne circule dans R_L. La sortie S prend le potentiel V_{CC}.
2. a saturé et b bloqué : Un courant circule dans R_L et le point S a un potentiel égal à V_{CEsat} de a, environ 0 V.
3. a bloqué et b saturé : Symétrique du précédent : S à 0 V.
4. a et b saturés : Un courant circule dans R_L et le potentiel de S s'établit à V_{CEsat} soit 0 V.

Calculons la fonction logique réalisée par la connexion des sorties, en utilisant la convention logique positive.

La table de vérité en logique positive donne : point commun = ab ce qui justifie le nom de ET câblé. En logique négative, on obtient un OU câblé. La démonstration précédente s'applique évidemment au-delà de deux variables.

a seul	b seul	point commun
OV	OV	OV
OV	5V	OV
5V	OV	OV
5V	5V	5V

Table de vérité électrique

a seul	b seul	point commun
0	0	0
0	1	0
1	0	0
1	1	1

Table de verité en logique positive

Fig. 4.11. — *Tables de vérité du point câblé en commun.*

6.3. Transmission d'une information parmi n

Réaliser la fonction ET sans employer de portes n'est pas le souci majeur d'un logicien. Par contre, il est souvent limité dans le nombre de fils pour transmettre de l'information. Il utilise alors le même fil pour transmettre une donnée à la fois parmi les *n* possibles. La figure 4.12 représente trois sources *a*, *b*, *c*, émettant sur un fil commun. Les trois sources sont connectées en ET câblé, mais seules quelques combinaisons de l'état de ces sources sont possibles :

— L'état inactif avec toutes les sources à l'état 1.

— L'état actif d'une source (0 logique), toutes les autres étant à l'état inactif (1 logique).

a	b	c	Utilisation
0	0	0	non utilisé
0	0	1	non utilisé
0	1	0	non utilisé
0	1	1	Source a active
1	0	0	non utilisé
1	0	1	Source b active
1	1	0	Source c active
1	1	1	état de repos

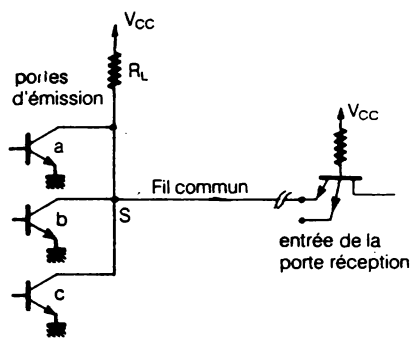


Fig. 4.12. — *Transmission d'information avec ET câblé.*

Avec ce dispositif, on transmet l'information sous la forme inverse puisque le zéro correspond à la source active. Le ET câblé avec collecteur ouvert pour cette fonction présente cependant quelques inconvénients :

a) La mise en parallèle des sorties additionne les capacités parasites de sorties de transistors ce qui donne de mauvaises performances de commutation, surtout quand une sortie passe de l'état 0 à l'état 1 (transistor bloqué). Il y a une importante dissymétrie dans les constantes de temps : $R_s.C_p$ avec R_s résistance de saturation (quelques dizaines d'ohms) quand on passe du niveau 1 au niveau 0 (front de descente raide), et $R_L C_p$ avec R_L résistance de charge très supérieure à Rs, ce qui donne un front de montée lent quand on passe du niveau 0 au niveau 1 (blocage). La figure 4.13 montre la déformation des signaux qui en résulte pour une porte NAND à collecteur ouvert.

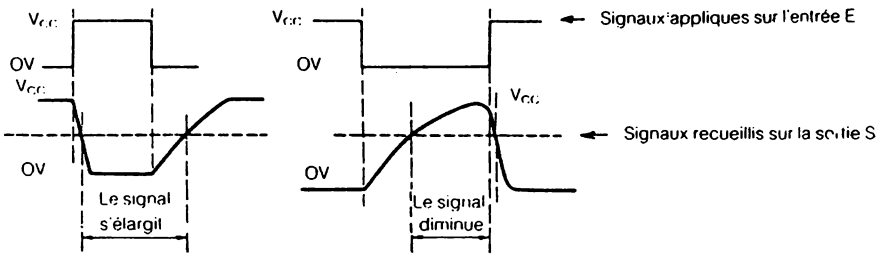


Fig. 4.13. — *Déformation des signaux liée aux charges capacitives.*

b) La mise en parallèle rend le dépannage difficile quand deux éléments ou plus sont en court-circuit à la masse. On ne sait pas localiser le défaut sans déconnecter une à une les portes.

c) Il faut câbler une résistance, ce qui peut introduire un élément étranger aux boîtiers logiques. Ceci n'est toutefois plus très gênant et il existe maintenant des résistances en boîtiers.

7. PORTE A SORTIE 3 ÉTATS "TRI-STATE"

Pour éviter les dissymétries de temps de commutation quand on utilise des portes à collecteur ouvert pour convoyer des informations d'origine multiple sur des fils communs, les constructeurs ont conçu des circuits dits

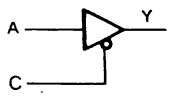
"tri-state" ou trois états que l'on peut réunir par leurs sorties. Les trois états du circuits sont :

- "1" (basse impédance)
- "0" (basse impédance)

} Comme pour une porte classique

- Déconnecté (très haute impédance).

En plus des entrées normales, une porte trois états (tri-state) possède une entrée de sélection qui connecte ou déconnecte la sortie. A titre d'exemple, nous donnons le tableau de fonctionnement du circuit "tri-state" 74125. Ce circuit possède une entrée A et une sortie Y, ainsi qu'une entrée de validation (ou de contrôle *C*). Les montages doivent être tels que la sortie d'un trois état ne soit pas utilisée lorsqu'elle se trouve dans l'état haute impédance.



C	A	Y	impédance de sortie
0 0	0 1	0 1	faible
1 1	0 1	Z	forte

Fig. 4.14. — *Circuit trois états 74LS125.*

L'état haute impédance de la porte s'obtient en modifiant le circuit commandant le totem-pôle de sortie. La commande *C* active (0 dans le cas du 74LS125) permet le fonctionnement normal des deux transistors du totem-pôle. Quand la commande *C* devient inactive, les deux transistors du totem-pôle se bloquent, ce qui donne une grande impédance de sortie.

A un instant donné, il ne doit y avoir qu'une seule sortie active à la fois, dans l'état 0 ou 1, les autres étant dans l'état haute impédance. Si deux sorties étaient actives en même temps, l'une à 0 l'autre à 1, il y aurait un court-circuit entre les deux, ce qui est néfaste et peut amener la destruction des circuits.

8. PORTE A ENTRÉES TRIGGER DE SCHMITT

Considérons un signal E attaquant un inverseur ou une entrée de porte. Pour obtenir un fonctionnement convenable, les fronts de montée et de descente du signal doivent rester dans certaines limites dépendant de la technologie utilisée (fig. 4.15).

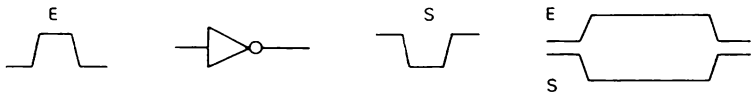


Fig. 4.15. – *Réponses d'un inverseur sur un signal à fronts raides.*

Si les fronts du signal d'entrée E ne sont pas suffisamment brefs, le signal de sortie S risque d'être perturbé. En effet l'entrée de l'inverseur se trouve pendant un temps relativement long dans la zone délimitée par les tensions V_a et V_b qui déterminent le seuil de basculement du circuit du niveau 1 vers le niveau zéro ou inversement. La sortie peut alors entrer en oscillation (fig. 4.16).

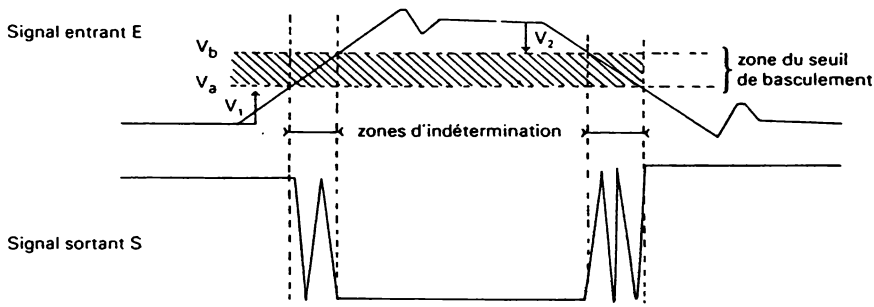


Fig. 4.16. – *Perturbation d'un signal sortant S sous l'influence d'un signal entrant E à front de transition lent.*

Pour pallier à ce défaut les constructeurs ont réalisés des circuits logiques, portes ou inverseurs dits à trigger de Schmitt qui présentent en entrée deux zones de basculement distinctes Z_1 et Z_2 correspondant respectivement aux seuils de passage 0 vers 1 et 1 vers zéro de la zortie (fig. 4.17).

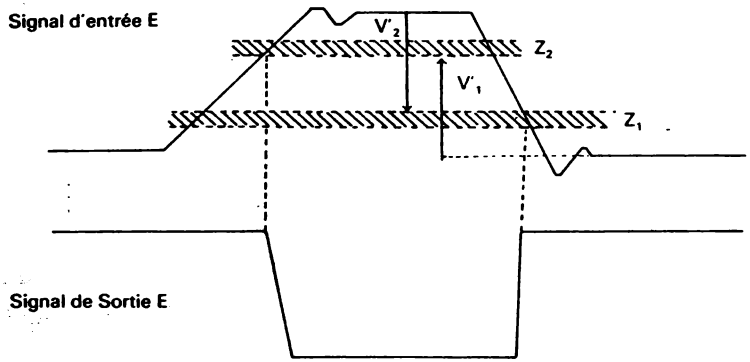


Fig. 4.17. — *Réponse d'un inverseur à trigger de Schmitt sous l'influence d'un signal de commande à front lent.*

Ces circuits à trigger de Schmitt présentent deux avantages. Premièrement les zones de basculement Z_1 et Z_2 étant distinctes, il ne peut y avoir d'oscillation lorsque le signal d'entrée reste dans l'une des zones même pendant un temps prolongé. Deuxièmement l'immunité aux bruits (tensions parasites sur les entrées) est supérieure. Elle est définie par V'_1 (fig. 4.17) au lieu de V_1 (fig. 4.16) pour le passage 1 vers 0 de la sortie et V'_2 (fig. 4.17) au lieu de V_2 (fig. 4.16) pour le passage 0 vers 1 de la sortie.

Les circuits trigger de Schmitt se distinguent par un symbolisme de représentation dont un exemple est donné à la figure 4.18 :

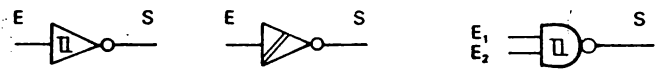


Fig. 4.18. — *Représentation symbolique de circuits à trigger de Schmitt.*

La figure 4.19 montre la réaction d'un inverseur trigger de Schmitt à un signal d'entrée fortement perturbé.

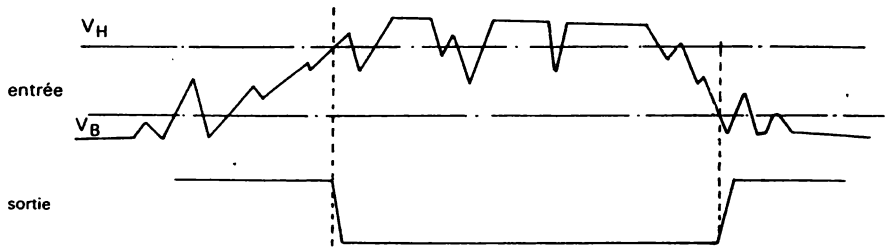


Fig. 4.19. — *Réponse d'un inverseur trigger de Schmitt à un signal fortement perturbé.*

Tant que le signal d'entrée n'atteint pas le seuil haut, la sortie reste à 1, puis elle bascule à 0 dès que le seuil haut est franchi. Ensuite, même si le signal d'entrée varie autour du seuil haut, la sortie demeure inchangée. Il faut attendre que le signal d'entrée redescende au-dessous du seuil bas pour que la sortie revienne à 1. De même, des variations autour du seuil bas, si elles n'atteignent pas le seuil haut ne donnent pas de variations en sortie. L'hystérésis V_H-V_B assure ainsi une certaine immunité au bruit.

Chapitre 5

LES PARAMÈTRES ÉLECTRIQUES ET TEMPORELS DES CIRCUITS LOGIQUES

1. INTRODUCTION

Dans le chapitre précédent, nous avons présenté un schéma de porte standard TTL et examiné son fonctionnement. Cette étude était en fait très sommaire et ne tenait pas compte de ce qui se passe pour une porte placée dans un contexte réel, quand elle-même distribue ses signaux de sortie vers d'autres circuits ou reçoit des signaux de ceux-ci. En particulier, rien n'avait été dit sur les phénomènes transitoires pourtant fondamentaux.

Plutôt que d'aller plus avant dans la voie de l'analyse du fonctionnement interne, nous allons nous contenter d'examiner les conséquences d'une telle analyse faite par les constructeurs et voir quelles en sont les contraintes pour l'utilisation des circuits. Ces contraintes sont définies par un ensemble de paramètres, les uns concernent les propriétés électriques de la porte et les autres les propriétés temporelles. Les circuits TTL-LS (Low Power Schottky) nous serviront d'exemple.

2. LA TENSION D'ALIMENTATION ET LA TEMPÉRATURE DE FONCTIONNEMENT

En logique TTL, la tension d'alimentation doit être bien stabilisée. Pour la gamme civile, travaillant de 0° C à 70° C, la tension nominale de + 5 *V* ne peut varier que de ± 5 % (4,75 *V* à 5,25 *V*) et pour la gamme militaire, la tolérance s'élargit à ± 10 % (4,5 *V* à 5,5 *V*). Il faut aussi que l'alimentation puisse accepter des appels brusques de courant. Les pointes de courant se produisent quand plusieurs circuits changent d'état en même temps. L'alimentation doit posséder une très bonne régulation de tension pour répondre aux appels de courant. De toute façon, elle ne pourra pas suivre complètement les variations instantanées. Pour suppléer à cette relative lenteur, on dispose au voisinage des boîtiers des condensateurs qui jouent un rôle de filtre et de réservoir en empêchant des variations trop rapides de la tension d'alimentation, en éliminant les pointes de tension, et en fournissant au circuit le courant instantané nécessaire au fonctionnement.

3. NIVEAUX DE SORTIE ET D'ENTRÉE D'UNE PORTE

Dans la présentation simpliste du chapitre précédent, nous supposions qu'une porte alimentée entre 0 *V* et + 5 *V* donnait des niveaux de sortie de 0 *V* pour le "0" et de + 5 *V* pour le "1" (logique positive). En fait, la porte délivre un signal compris respectivement entre 0 *V* et une valeur maximum pour le "0" logique et entre + 5 *V* et une valeur minimum pour le "1" logique. Le constructeur garantit pour les portes de la famille un niveau $V_{OL\ max}$ (Voltage Output Low : tension de sortie au niveau bas) tel qu'aucune porte ne délivrera un niveau de sortie supérieur à $V_{OL\ max}$ et il garantit également un niveau $V_{OH\ min}$ (Voltage Output High : tension de sortie au niveau haut) tel qu'aucune porte ne délivrera un niveau de sortie inférieur à $V_{OH\ min}$.

Ces niveaux V_{OL}, V_{OH} sont reçus par des entrées de porte qui doivent avoir une plage de reconnaissance du "0" et du "1" plus large que la plage $0\ V - V_{OL\ max}$ et $+\ 5\ V - V_{OH\ min}$. C'est pourquoi la porte est caractérisée par un seuil $V_{IL\ max}$ (Voltage Input Low : seuil d'entrée au niveau bas) plus grand que $V_{OL\ max}$ et par un seuil $V_{OH\ min}$ (Voltage Input High : seuil d'entrée au niveau haut) plus petit que $V_{OH\ min}$.

L'immunité au bruit (fig. 5.1) se caractérise par l'amplitude du signal parasite entrant qui provoque un changement d'état à la sortie du circuit. Si l'on prend le pire des cas, l'immunité au bruit est de $V_{OH} - V_{IH}$ sur le niveau 1 et de $V_{IL} - V_{OL}$ sur le niveau 0. Il s'agit de l'immunité statique. En pratique, on dépasse ces valeurs si le signal parasite est suffisamment bref, de l'ordre du retard d'une porte.

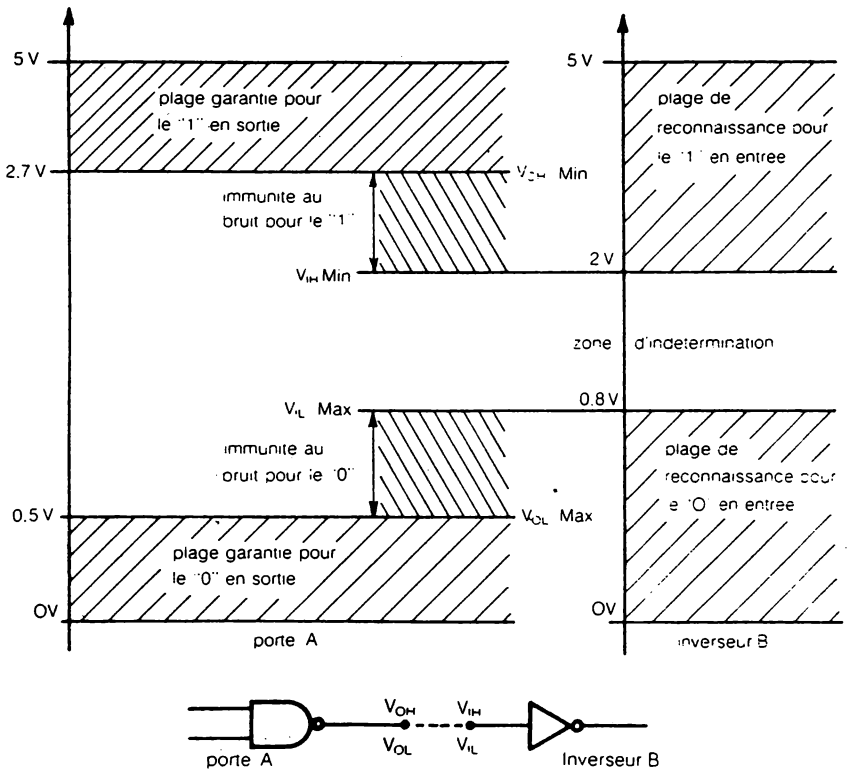


Fig. 5.1. — *Connexion entre deux portes et tensions correspondantes pour des circuits TTL-LS.*

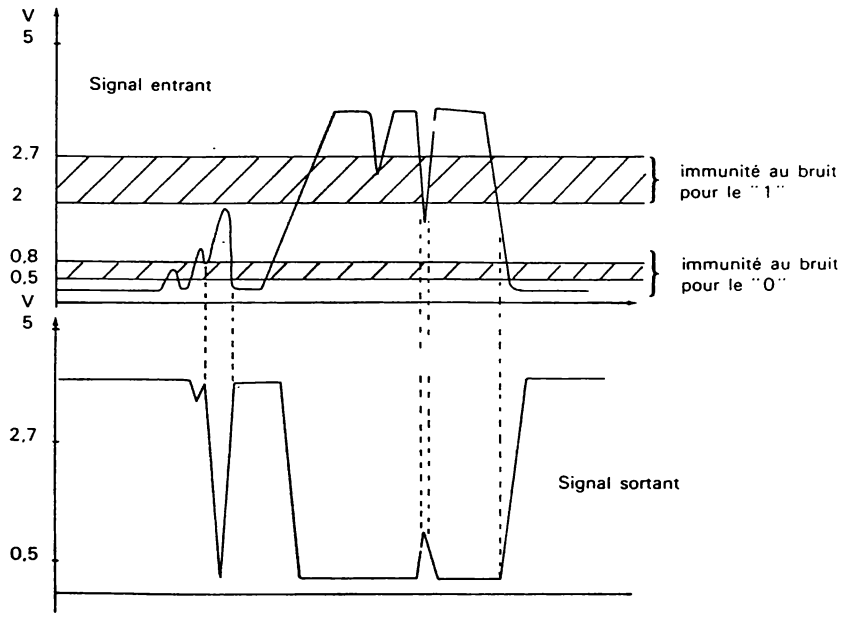


Fig. 5.2. — *Immunité au bruit.*

La figure 5.2 montre un signal parasité traversant l'inverseur *B*.

4. COURANTS DE SORTIE OU D'ENTRÉE D'UNE PORTE STANDARD (SORTIE TOTEM-POLE)

Lorsqu'une sortie de porte TTL est au niveau bas V_{OL}, elle absorbe un courant noté I_{OL} (Low Level Output Current : courant de sortie au niveau bas), courant fourni par l'entrée de la porte ou les entrées des portes auxquelles la première est raccordée. Ces courants d'entrée fournis sont appelés I_{IL} (Low Level Input Current : courant d'entrée au niveau bas).

A l'opposé, une sortie de porte TTL au niveau haut fournit un courant I_{OH} (High Level Output Current : courant de sortie au niveau haut), courant absorbé par les entrées auxquelles la sortie est raccordée. Ces courants d'entrée sont appelés I_{IH} (High Level Input Current : courant d'entrée au niveau haut).

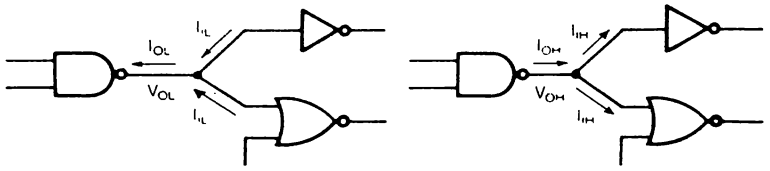


Fig. 5.3. – *Sens des courants fournis et absorbés.*

Le courant de sortie est la somme des courants d'entrée.

$I_{OH} = \sum I_{IH}$ et $I_{OL} = \sum I_{IL}$

La valeur des courants I_{OH}, I_{OL}, I_{IH}, I_{IL} est caractéristique de la famille. Pour la famille Low Power Schottky, les valeurs sont :

$I_{OL} = + 8\ mA$ $\qquad$ $I_{IL} = - 0{,}4\ mA$

$I_{OH} = - 400\ \mu A$ $\qquad$ $I_{IH} = + 20\ \mu A$

Le signe + signifie que la porte absorbe du courant, et le signe – qu'elle en émet. Les courants aux niveaux bas et haut sont très différents. Les premiers correspondent à des courants directs et les seconds à des courants inverses.

5. COURANT DE SORTIE POUR UNE PORTE A COLLECTEUR OUVERT (OPEN COLLECTOR) ET TROIS ÉTATS (TRI-STATE)

Les entrées des portes à collecteur ouvert et trois états ne présentent pas de particularité par rapport aux portes standard. Par contre, les caractéristiques des sorties diffèrent.

5.1. Sortie à collecteur ouvert

Une sortie de porte standard, à sortie "totem pôle" absorbe du courant au niveau bas et en fournit au niveau haut. Une sortie de porte collecteur ouvert absorbe également du courant au niveau bas mais est bien sûr incapable d'en fournir quand elle est bloquée. Par contre, la jonction base collec-

teur absorbe un léger courant inverse quand le collecteur est porté au niveau haut. Ainsi pour la porte NAND à collecteur ouvert de type 74 LS 01, on a :

$I_{OL} = + 8\ mA$ $\qquad$ $I_{OH} = + 100\ \mu A$

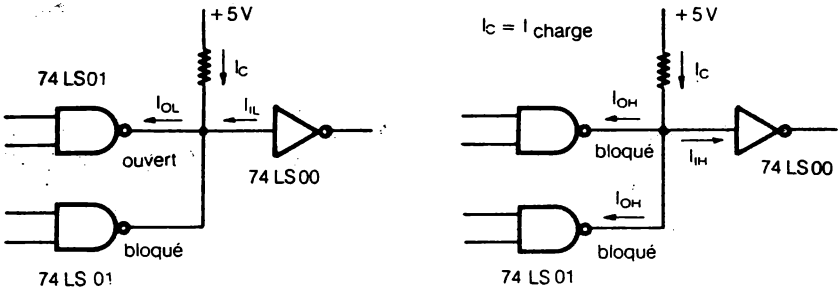


Fig. 5.4. — *Connexion de portes à collecteur ouvert.*

La figure 5.4 montre que la sortie d'une porte bloquée se comporte au niveau haut comme une entrée de porte normale.

5.2. Sortie trois états

Quand une sortie "tri-state" est activée, elle se comporte comme une sortie "totem pôle" normale, avec des courants I_{OH} et I_{OL}. La seule différence réside dans la valeur de ces courants. Le courant I_{OH}, très faible pour une porte standard, est élevé dans une porte "tri-state" afin d'améliorer la vitesse de commutation au niveau haut.

74 LS 00 : $I_{OH} = -400\ \mu A$, $I_{OL} = 8\ mA$ (porte standard)

74 LS 125 : $I_{OH} = -2{,}6\ mA$, $I_{OL} = 16\ mA$ (porte "tri-state")

Par contre, lorsque l'on désactive la sortie "tri-state", les deux transistors du "totem-pôle" sont bloqués. Le potentiel de sortie est fixé par celui de la porte connectée en parallèle, et active au niveau 1 ou 0. La porte à l'état haute impédance absorbe ou fournit un courant I_{OZ} (Off State ou High Impedance Output Current : courant à l'état haute impédance).

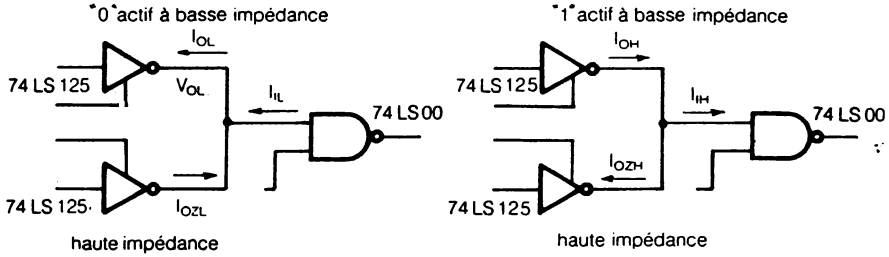


Fig. 5.5. – *Connexion de portes tri-state.*

Ces deux courants sont notés I_{OZH} et I_{OZL} selon que la porte active qui impose le niveau est au niveau haut ou bas. La figure 5.5. montre qu'une sortie de porte à l'état haute impédance se comporte comme une entrée de porte normale. Dans les deux cas, les courants sont faibles. Ils correspondent en valeur absolue au courant absorbé par l'entrée d'une porte au niveau haut :

74 LS 125 : $I_{OZH} = 20\ \mu A, I_{OZL} = -20\ \mu A$ (Porte trois états)

74 LS 00 : $I_{IH} = 20\ \mu A$ (Porte NAND)

6. PARAMÈTRES D'ENTRÉES D'UNE PORTE "TRIGGER DE SCHMITT"

Nous avons vu dans le chapitre précédent, le principe des portes "Trigger de Schmitt". En fonction des particularités d'entrée de ces portes, des paramètres supplémentaires sont définis :

V_T+ : Seuil de basculement haut.

V_T- : Seuil de basculement bas.

I_T+ : Courant d'entrée au seuil haut.

I_T- : Courant d'entrée au seuil bas.

V_T+ et V_T- remplacent V_{IH} et V_{IL}. En plus de I_T+ et I_T-, le constructeur donne les courants d'entrée I_{IH} et I_{IL} lorsque le signal d'entrée provient d'une sortie de porte fournissant V_{OH} et V_{OL}.

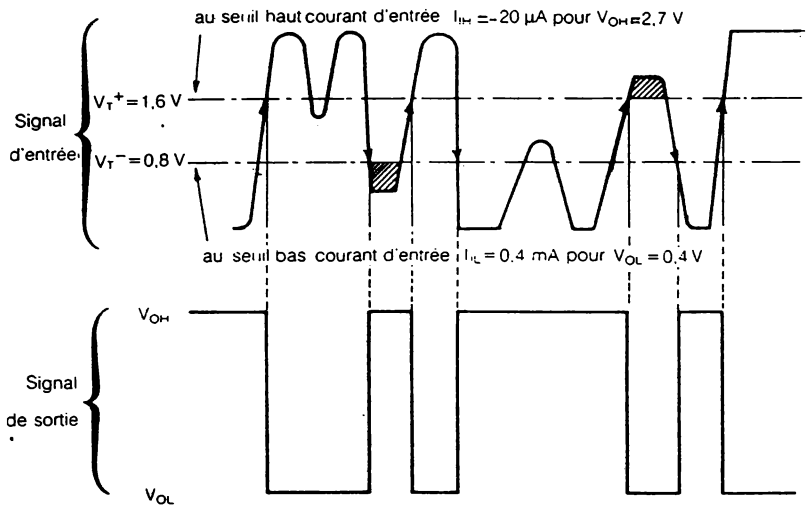


Fig. 5.6. — *Porte trigger de Schmitt 74 LS 14 : paramètres d'entrée.*

7. AUTRES PARAMÈTRES ÉLECTRIQUES

Dans ce qui précède, nous avons défini les paramètres électriques essentiels. Il en existe encore quelques autres. Citons-les en expliquant leur rôle.

7.1. Tension et courant d'écrêtage

La porte TTL comporte outre les diodes normales ou les émetteurs multiples d'entrées, des diodes servant à l'écrêtage des tensions négatives. On définit une tension d'écrêtage et le courant correspondant :

V_{IK} : "Voltage input clamping"

I_I : "Input current" au moment de l'écrêtage (clamping).

Les figures 5.7 et 5.8 illustrent l'effet d'une diode d'écrêtage sur un signal d'entrée présentant un potentiel négatif.

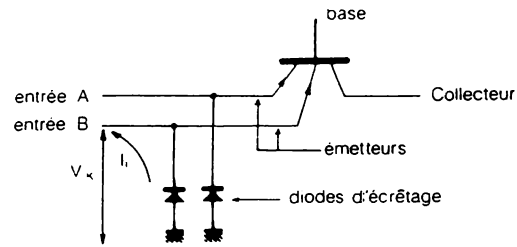


Fig. 5.7. – *Entrée d'une porte comportant des diodes d'écrêtage.*

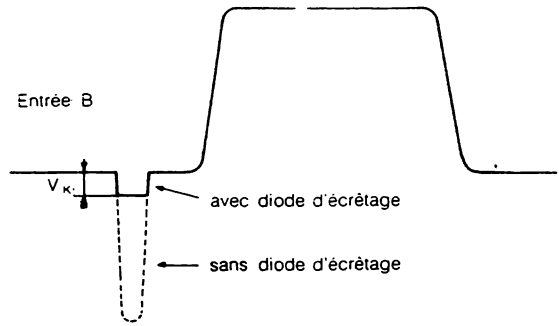


Fig. 5.8. – *Effet d'une diode d'écrêtage sur un signal appliqué sur l'entrée d'une porte.*

7.2 Courant d'entrée à la tension d'entrée maximum

Ce courant noté I_I est défini à la tension maximum supportable par l'entrée. Par exemple, avec la série TTL standard, une entrée peut être portée à + 5,5 *V* et absorbe alors un courant de + 1 *mA*, ce qui est énorme vis-à-vis du courant normalement absorbé I_{IH} de + 40 μA. Pour éviter la dissipation correspondante, il ne faut pas mettre une entrée de TTL standard au + 5 *V* car la moindre surtension excédant 5,5 *V* est dangereuse. On doit polariser cette entrée au + 5 *V* par l'intermédiaire d'une résistance si elle n'est pas reliée normalement à la sortie d'une porte logique.

Par contre, en technologie LS, l'entrée supporte + 7 *V* et le courant absorbé n'est que 0,1 *mA*. Ceci explique que l'on puisse polariser directement une entrée non utilisée au + 5 *V* de l'alimentation.

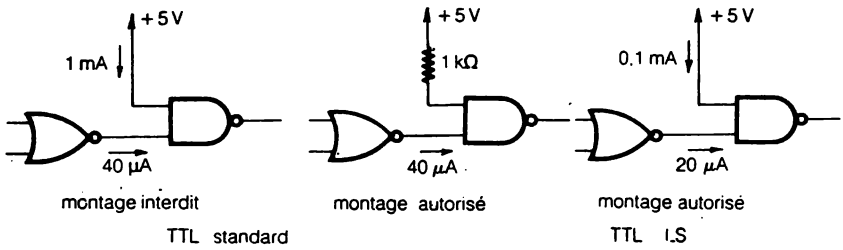


Fig. 5.9. — *Polarisation d'entrées non utilisées.*

7.3. Courant de court-circuit

C'est le courant appelé I_{OS} (Short Circuit Output Current) qui émane d'une sortie normalement à l'état haut V_{OH}, d'après les conditions logiques appliquées aux entrées et mise à la masse par un court-circuit. Ce courant fourni est très important (— 100 mA pour une porte TTL-LS) alors que la porte fournit normalement dans cet état le courant I_{OH}, beaucoup plus faible ($I_{OH} = -400\ \mu A$). Il est recommandé de ne pas mettre plus d'une sortie en court-circuit à la fois dans un boîtier (sinon la puissance dissipée abîme le boîtier) et pour certaines familles dont la LS, le court-circuit ne doit pas durer plus d'une seconde.

Pourquoi spécifier cette grandeur ? Quand il y a des court-circuits accidentels, ils ne respectent évidemment pas les contraintes du constructeur. Par contre, le réalisateur d'un système peut effectuer des court-circuits volontaires pour imposer lors d'une mise au point, une sortie à l'état 0 alors qu'elle se trouvait normalement à l'état 1. De même, lors du contrôle automatique des montages par un système de test, le calculateur peut imposer un niveau 0 sur une broche indépendamment du fonctionnement normal. Il est bon alors que le programmeur sache que le court-circuit ne doit pas durer plus d'une seconde, sinon il détruirait ce qu'il a la charge de dépanner.

La liaison directe d'une sortie de porte au + 5 V est interdite, elle a pour effet de détruire le circuit, si sa sortie est au niveau bas, par la mise en court-circuit de l'un des transistors du totem pôle.

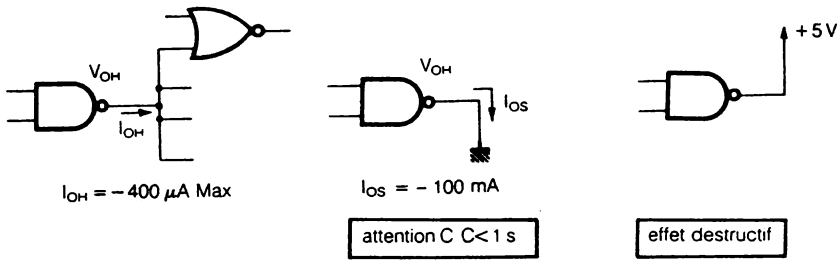


Fig. 5.10. — *Courant de court-circuit pour les portes TTL-LS.*

7.4. Courant d'alimentation et puissance consommée

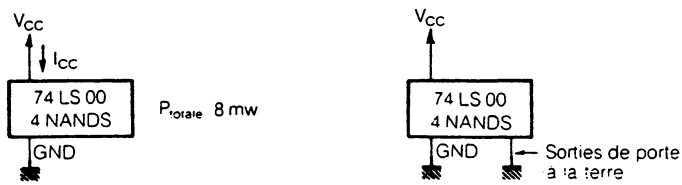
Le courant d'alimentation, noté I_{cc} permet de calculer l'énergie consommée par un boîtier. Il suffit de multiplier ce courant I_{cc} par la tension d'alimentation :

Exemple :

74 LS 00 : $I_{cc\ moyen}$ par porte $= 0{,}4\ mA$ donc $P = 2\ mW$

74 LS 125 : $I_{cc\ typique}$ par boîtier $= 11\ mA$ donc $P = 55\ mW$

Pour une porte CMOS la puissance consommée est proportionnelle à la fréquence de commutation.



Nombre de sorties à la terre	Puissance consommée	Effet sur la durée de vie
0	8 mw	Aucun
1	500 mw	Aucun si < 1 s
2	1000 mw	danger destructif
3	1500 mw	
4	2000 mw	

Fig. 5.11. — *Puissance consommée par un boîtier.*

8. PARAMÈTRES TEMPORELS DES CIRCUITS COMBINATOIRES ET SÉQUENTIELS

On appelle combinatoire un circuit dont l'état permanent de sortie ne dépend que de l'état permanent des entrées, et pas du tout de l'ordre chronologique dans lequel les entrées ont atteint leur état permanent. Les portes logiques étudiées jusqu'à présent rentrent dans cette catégorie.

On appelle séquentiel un circuit dont l'état permanent de sortie dépend de l'ordre dans lequel les niveaux d'entrée ont été appliqués. On dit qu'un circuit séquentiel conserve la mémoire des événements présentés aux entrées.

En dehors des états permanents, les sorties des circuits combinatoires ou séquentiels présentent des états transitoires, de courte durée, au moment où les états logiques de une ou plusieurs entrées varient.

Les paramètres temporels définissent, pour un circuit combinatoire, les retards de commutation et pour un circuit séquentiel, les retards de commutation ainsi que les contraintes d'échelonnement des signaux pour que le circuit fonctionne correctement.

9. CARACTÉRISATION D'UNE IMPULSION

Pour caractériser une impulsion (fig. 5.12), on définit :

- La durée de l'impulsion notée t_W : (Width : largeur) au niveau haut et bas.
- Le front de montée t_R (t_{rise}), mesuré entre le niveau bas et le niveau haut à 10 % et 90 %.
- Le front de descente t_F (t_{fall}), mesuré dans les mêmes conditions.
- La fréquence de répétition du signal (inverse de la période T).

En logique TTL, on ne trouve pas de spécification des valeurs des fronts de montée et descente. Pour une porte logique LS, chargée par 50 *pF*, ils sont de 6 à 8 *ns* pour le front de descente, 9 à 12 *ns* pour le front de montée. Pour les portes non plus, la durée minimale d'une impulsion n'est pas spécifiée. Elle l'est, par contre, pour les circuits séquentiels. De même, la fréquence maximum de répétition n'est spécifiée que pour les circuits séquentiels.

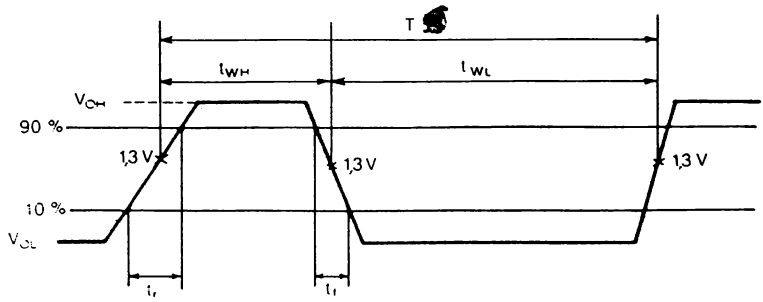


Fig. 5.12. – *Paramètres d'une impulsion.*

10. TEMPS DE PROPAGATION

Lorsqu'un signal traverse une porte ou un assemblage de portes ou plus généralement lorsqu'un signal agit sur un circuit, l'effet de cette action se manifeste avec retard sur la sortie. Le retard n'est pas le même pour le front de montée ou de descente. La figure 5.13 montre les deux retards notés t_{PLH} (propagation delay time low to high level : temps de propagation du niveau bas au niveau haut) et t_{PHL} (propagation delay time high to low level : temps de propagation du niveau haut au niveau bas), après traversée d'un inverseur. Pour la porte 74 LS 00, on a : $t_{PLH} = t_{PHL} = 15$ *ns* en valeur maximum.

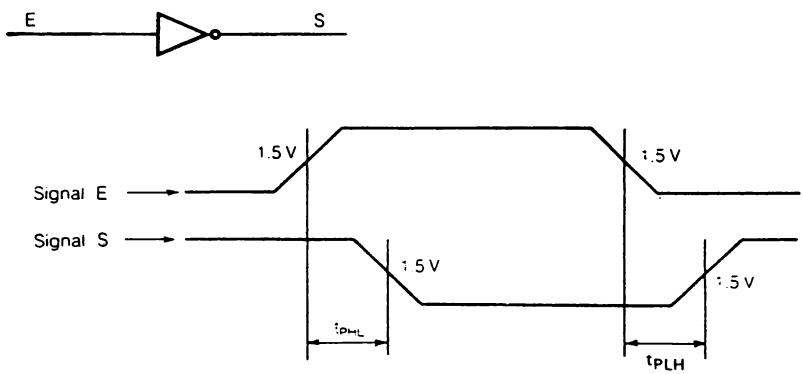


Fig. 5.13. – *Temps de retard d'une porte.*

11. TEMPS D'OUVERTURE ET DE FERMETURE D'UNE PORTE "TRI-STATE"

Quand une porte "tri-state" est activée, elle se comporte du point de vue de retards comme une porte à sortie "totem-pôle". Par contre, de nouveaux paramètres définissent le passage de l'état bloqué à l'état actif et inversement. On obtient ainsi 4 paramètres :

t_{PZH} : Temps de validation à l'état haut (Output enable time to high level).

t_{PZL} : Temps de validation à l'état bas (Output enable time to low level).

t_{PHZ} : Temps de désactivation de l'état haut (Output disable time from high level).

t_{PLZ} : Temps de désactivation de l'état bas (Output disable time from low level).

La figure 5.14 montre ces paramètres :

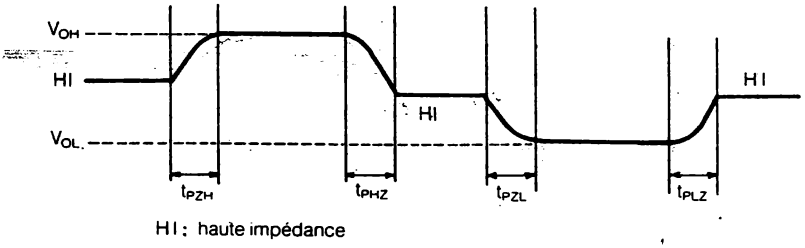


Fig. 5.14. — *Paramètres d'activation et de désactivation d'une porte "tri-state".*

Pour la porte tri-state 74 LS 125, on a :

$t_{PZH} = 17$ ns, $t_{PZL} = 25$ *ns*, $t_{PHZ} = 8$ *ns*, $t_{PLZ} = 12$ *ns*.

12. PARAMÈTRES TEMPORELS SPÉCIFIQUES AUX CIRCUITS SÉQUENTIELS

Tous les paramètres précédents s'appliquent aussi bien aux circuits combinatoires que séquentiels. Les paramètres suivants ne concernent que

les circuits séquentiels. Ils expriment la relation de temps que doivent respecter deux signaux l'un par rapport à l'autre pour que le circuit fonctionne correctement. La figure 5.15 montre ces paramètres.

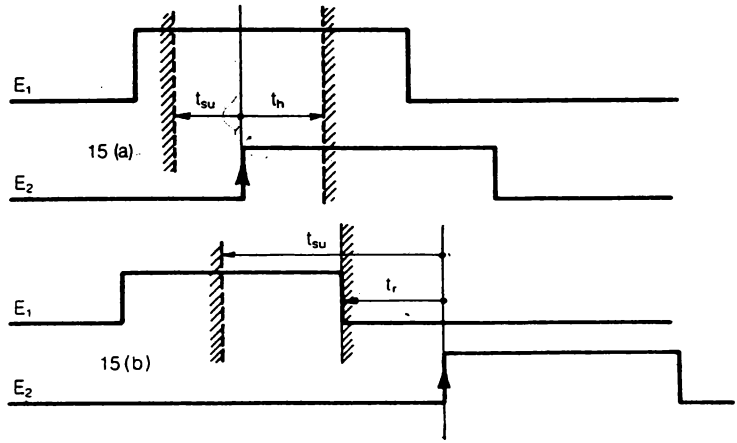


Fig. 5.15. — *Définition des paramètres t_{su}, t_h et t_r.*

Par exemple, un circuit séquentiel fonctionne avec deux signaux E_1 et E_2. La figure 5.15 a définit les paramètres t_{su} ou $t_{set\ up}$ (temps de préaffichage) et t_h ou t_{hold} (temps de maintien). La figure 5.15b explicite le paramètre t_r ou $t_{release}$ (temps de relâchement). Le temps de préaffichage t_{su} traduit qu'il faut que E_1 soit établi de manière stable à 0 ou 1 un certain temps *avant* le front montant de E_2, et le temps de maintien indique que E_1 doit être stable à 0 ou 1 un certain temps *après* le front montant de E_2.

La figure 5.15b, outre le t_{su}, montre le t_r qui indique que l'on peut enlever le signal E_1 un certain temps *avant* le front montant de E_2. On le définit également comme un t_{hold} négatif.

Ces paramètres sont très importants pour les circuits séquentiels et nous les reverrons plus loin en étudiant ces types de circuits.

13. UTILISATION DES PARAMÈTRES ÉLECTRIQUES ET TEMPORELS

Maintenant que les paramètres électriques et temporels sont définis, examinons l'usage que peut en faire le concepteur de système.

13.1. Paramètres V_{OH}, V_{OL}, I_{OH}, I_{OL}, V_{IH}, V_{IL}, I_{IH}, I_{IL}

13.1.1. Entrance et sortance d'une porte (fan in, fan out)

Lorsque l'on raccorde des portes logiques de la même famille entre elles, les paramètres de tension de sortie V_{OH}, V_{OL}, et les seuils d'entrée V_{IH} et V_{IL} sont évidemment compatibles et l'utilisateur n'a pas à s'en soucier. En ce qui concerne les courants d'entrée et de sortie, on ne compte généralement pas en μA ou mA mais en "charges".

Une entrée de circuit représente une charge si elle consomme ou fournit le courant de base de la porte de référence (la porte NAND en TTL). Ainsi en TTL-LS une charge correspond à :

$I_{IH} = 20\ \mu A$ $\qquad$ $I_{IL} = -0{,}4\ mA$

Si une entrée absorbe $40\ \mu A$ et fournit $-0{,}8\ mA$, elle compte pour 2 charges, et ainsi de suite.

Le courant maximum que peut fournir ou absorber une sortie est lui aussi compté en charges. Il est égal au plus petit des rapports :

$$\frac{I_{OH}}{I_{IH}} \text{ et } \frac{I_{OL}}{I_{IL}}$$

Ainsi, pour la porte de base 74 LS 00, on a :

$I_{OH} = -400\ \mu A$, $I_{IH} = 20\ \mu A$, $I_{OL} = 8\ mA$, $I_{IL} = -0{,}4\ mA$

D'où un rapport :

$$\frac{I_{OH}}{I_{IH}} = \frac{400}{20} = 20 \qquad \frac{I_{OL}}{I_{IL}} = \frac{8}{0{,}4} = 20$$

Une porte standard en TTL-LS peut donc alimenter 20 charges. On dit que la sortie a une sortance de 20 (ou fan out) et que l'entrée a une entrance (ou fan in) de 1.

Pour la plupart des circuits d'usage courant, la connexion est très simple. Il suffit de compter le nombre d'entrées reliées à une sortie et faire en sorte de ne pas atteindre 20. Pour certains circuits, il faudra vérifier si la sortance et l'entrance sont bien de 20 et de 1, et en tenir compte dans le bilan. Cette règle des charges constitue une règle de base pour interconnecter les opérateurs logiques.

La figure 5.16 illustre sur quelques opérateurs TTL-LS les notions d'entrance et de sortance.

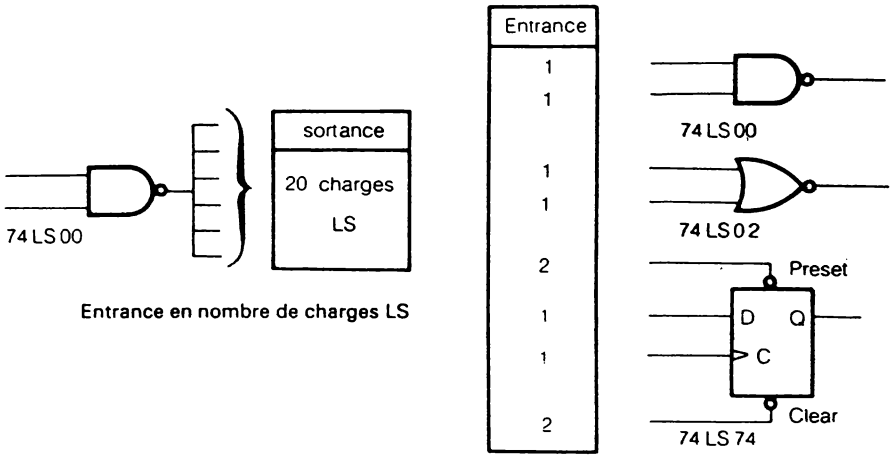


Fig. 5.16. — *Entrance et sortance de quelques opérateurs TTL-LS.*

13.1.2. Connexion entre circuits de familles différentes

Pour connecter ensemble deux circuits de familles logiques différentes, il faut :

- Que la plage de reconnaissance du 1 et du 0 (V_{IH} et V_{IL}) soit plus large que la plage de sortie du 1 et du 0 (V_{OH} et V_{OL}) pour assurer une certaine immunité au bruit.
- Que les courants de sortie I_{OH} et I_{OL} soient de même sens que les courants d'entrée I_{IH} et I_{IL} et supérieurs à ceux-ci.

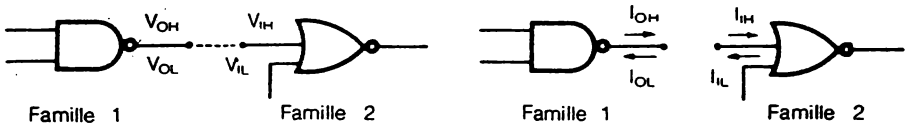


Fig. 5.17. — *Interface entre deux familles.*

- Que les fronts de montée et de descente de la sortie soient compatibles avec les entrées. En logique TTL, il ne faut pas que les fronts de montée et de descente soient trop lents, sinon, la sortie oscille quand le niveau d'entrée se situe dans la zone d'incertitude entre le $V_{IL\ max}$ et le $V_{IH\ min}$. En TTL-LS, les fronts de montée sont de l'ordre de 12 *ns* et les fronts de descente de l'ordre de 8 *ns*.

13.1.3. Connexion avec des produits compatibles TTL

Bien des circuits réalisés dans une technologie différente du bipolaire TTL (ECL, MOS, CMOS) présentent des entrées et sorties compatibles TTL. Cela signifie que les caractéristiques de tension V_{OH}, V_{OL}, V_{IH}, V_{IL} sont identiques ou compatibles. Par contre, les limitations se situent au niveau des courants. La sortance est généralement limitée à une ou quelques unités d'une famille donnée (TTL standard, TTL-LS ou TTL-S). L'entrance peut également être différente de l'entrance normale. Dans ce cas, il faut consulter les caractéristiques du produit et les comparer avec celles de la famille TTL.

13.2. Paramètres I_{OZH}, I_{OZL}, I_{OL}, V_{IK}, I_I, I_{OS}, I_{CC}

Les paramètres I_{OZH}, I_{OZL} servent surtout à titre d'information pour montrer qu'une porte tri-state bloquée absorbe ou émet très peu de courant et que l'on peut négliger les effets électriques d'une porte tri-state bloquée.

Le paramètre I_{OL} sert au calcul d'un assemblage de portes à collecteur ouvert.

Le paramètre V_{IK} sert lui aussi à titre d'information. On sait que les entrées sont protégées des tensions négatives, qui n'ont pas normalement leur place dans une famille TTL (il existe cependant des pointes négatives par suite d'oscillations).

Le paramètre I_I indique qu'il ne faut pas raccorder une entrée de porte TTL standard au + 5 V directement, mais que ce câblage est possible en TTL-LS.

Le paramètre de courant de court-circuit I_{OS} avertit l'utilisateur qu'il ne peut pas mettre inconsidérément les sorties en court-circuit pour la mise au point, la maintenance, ou le test sans dommage pour le circuit.

Le courant I_{CC} permet de faire des prévisions de consommation et de déterminer la puissance d'alimentation nécessaire.

13.3. Paramètres temporels de temps de propagation: t_{PHL}, t_{PLH}, t_{PHZ}, t_{PLZ}, t_{PZH}, t_{PZL}

13.3.1. Porte ordinaires (NAND, NOR ET, OU...)

Les paramètres t_{PLH} et t_{PHL} maximum sont très utiles au concepteur. Ils permettent de calculer le temps maximum d'établissement d'une fonction logique traversant une chaîne d'opérateurs en cascade. C'est seulement après ce temps maximum que la fonction peut être exploitée en toute sécurité. Les valeurs typiques de t_{PLH} et t_{PHL} ne sont données qu'à titre d'information.

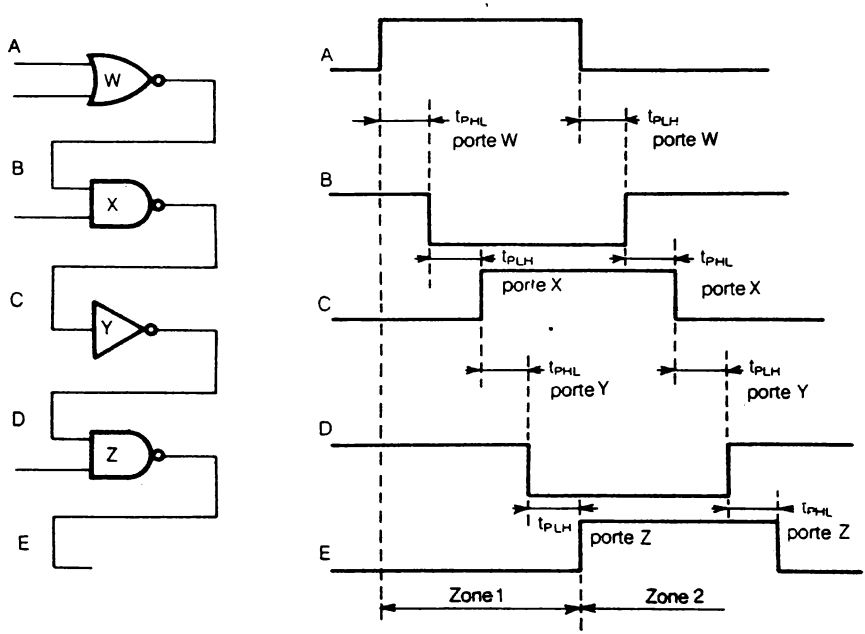


Fig. 5.18. — *Temps de transfert d'une information à travers une suite de portes.*

Le signal E de la figure 5.18 sortant de la porte Z ne peut être exploité de façon sûre avant le début de la zone 2. La zone 1 correspond à un intervalle de temps d'instabilité du signal. Dans les conditions d'alimentation de charge et de température définies au catalogue, le constructeur garantit la présence du signal à la sortie de la porte Z au bout d'un temps t tel que:

$$t = t_{PHL}\,(\text{porte } W) + t_{PLH}\,(\text{porte } X) + t_{PHL}\,(\text{porte } Y) + t_{PLH}\,(\text{porte } Z)$$

tous les temps étant pris à la valeur maximum.

13.3.2. Porte Tri-state

Les figures 5.19 et 5.20 représentent deux montagnes courants avec des portes trois états. Le premier correspond à un aiguillage de plusieurs informations avec utilisation de l'information transmise par un seul circuit (porte NAND) et le second à un aiguillage avec une diffusion vers plusieurs utilisateurs (liaison bus).

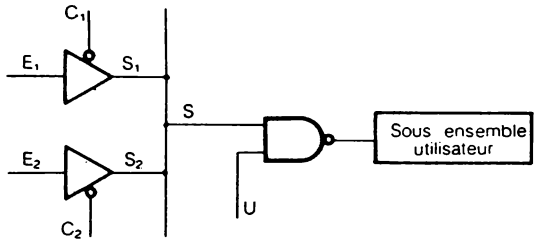


Fig. 5.19. — *Plusieurs circuits 3 états reliés par la sortie attaquent une porte logique.*

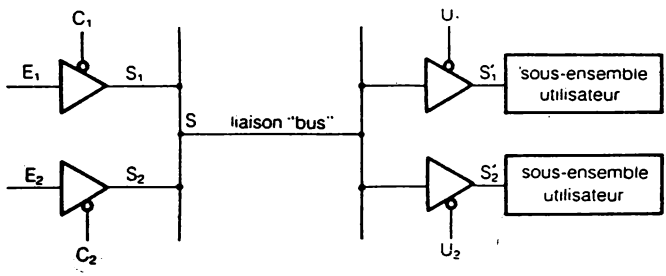


Fig. 5.20. — *Plusieurs circuits 3 états reliés par la sortie attaquent plusieurs entrées de circuit 3 états.*

Deux règles très importantes doivent être respectées dans l'utilisation des portes trois états :

— Il ne faut pas que plusieurs sorties soient simultanément en mode basse impédance.

— Il ne faut pas exploiter une sortie trois états au moment où elle se trouve en mode haute impédance (état logique non défini).

La figure 5.21 montre comment les paramètres t_{PHZ}, t_{PLZ}, t_{PZL} et t_{PZH} permettent de respecter la première règle. Si les commandes C_1 et C_2 sont trop proches dans le temps, la sortie S_1 peut ne pas être revenue à l'état

haute impédance alors que la sortie S_2 peut déjà être parvenue à l'état basse impédance. Il en résulte un court-circuit temporaire, comme le montre la figure 5.22, si une des sorties se trouve au niveau haut et l'autre au niveau bas. On dit alors qu'il y a *conflit de sorties* ou *conflit de bus (pour les liaisons bus). Si d* représente la durée séparant la fin de C_1 et le début de C_2, on doit satisfaire l'inéquation :

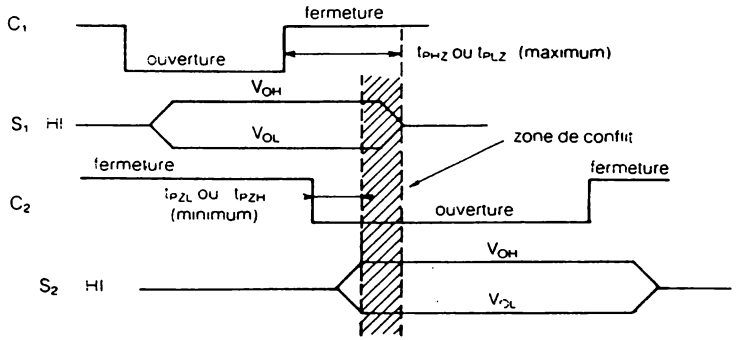


Fig. 5.21. — *Porte 3 états : respect des paramètres dynamiques.*

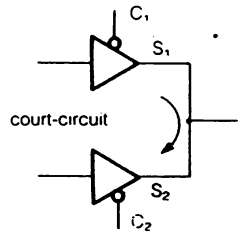


Fig. 5.22. — *Circuits 3 états : court circuit par les sorties.*

$$d + (t_{PZH} \text{ ou } t_{PZL})_{min} > (t_{PHZ} \text{ ou } t_{PLZ})_{max}$$

Comme les t_{PZH} ou t_{PZL} ne sont pas spécifiés en valeur minimum, cela devient :

$$d > (t_{PHZ} \text{ ou } t_{PLZ})_{max}$$

La figure 5.23 montre comment respecter la seconde règle. Il faut que l'impulsion d'utilisation U soit comprise dans la zone où la sortie sélectionnée par une commande se trouve dans l'état basse impédance. L'impulsion U doit donc commencer au plus tôt un temps t_{PZH} ou t_{PZL} maximum

après le début de C et doit se terminer avant un temps t_{PLZ} ou t_{PHZ} minimum suivant la fin de C. Comme les temps ne sont pas spécifiés au minimum, la fin de U doit coïncider au plus tard avec la fin de C. Dans la pratique, l'impulsion U couvre une zone plus restreinte, afin d'éviter les dispersions éventuelles.

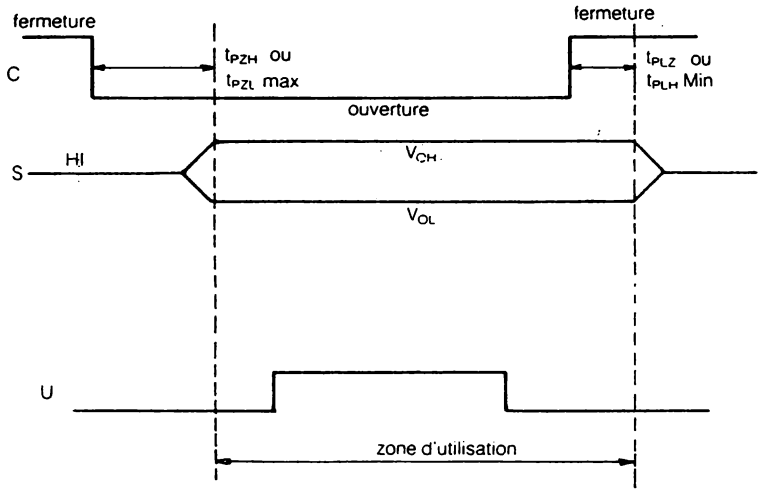


Fig. 5.23. — *Prélèvement d'une information sur une sortie 3 états.*

13.4. Paramètres temporels F, t_W, t_{su}, t_h, t_r

Le paramètre F, fréquence maximum n'est pas très utile au concepteur. Il permet simplement de situer les performances du produit. En effet, la fréquence maximum de fonctionnement d'un circuit seul et du même circuit plongé dans un environnement ne sont pas les mêmes, à cause des contraintes temporelles des circuits associés.

La durée t_W des impulsions est elle aussi indicative et généralement très petite, de quelques dizaines de *ns*. Pour une bonne fiabilité, il faut se garder d'employer des impulsions de durée voisine de t_W minimum, car ces impulsions sont facilement altérables et peuvent devenir inférieures aux t_W minimum.

Les paramètres t_{su}, t_h, t_r sont vitaux pour le fonctionnement des circuits séquentiels. Nous aurons l'occasion d'y revenir en étudiant les circuits séquentiels.

13.5. Variation des paramètres

Les paramètres sont mesurés dans des conditions précises de tension d'alimentation, de tension d'entrée, de capacité, de résistance de charge et de température. Les valeurs des paramètres sont données le plus souvent à 25° C et $V_{cc} = 5\ V$. Cependant ces valeurs, notamment les temps de propagation dépendent beaucoup de la charge capacitive (les valeurs courantes dans les spécifications sont de $15\ pF$, $25\ pF$, ou $50\ pF$), de la tension d'alimentation et de la température. Le concepteur a besoin de connaître les valeurs extrêmes de ces paramètres quand la tension varie de $\pm$ 5 % (4,75 V à 5,25 V) et quand la température varie de 0° C à 70° C pour la gamme commerciale ou quand la tension varie de $\pm$ 10 % (4,5 V à 5,5 V) et la température varie de $-$ 55° C à + 125° C pour la gamme militaire. Les catalogues de circuits intégrés ne donnent pas toujours de réponse satisfaisante à ces questions. On trouve parfois quelques courbes d'évolution du t_{PHL}, t_{PLH} en fonction de la température, de la charge capacitive ou de la tension d'alimentation. Ces renseignements concernent la porte de base mais on ne sait pas comment les extrapoler à d'autres circuits.

Un constructeur, Advanced Micro Devices (AMD) apporte cependant une réponse intéressante en donnant pour ses circuits trois types de spécifications. Les circuits 74 LS et Am 25 LS sont spécifiés normalement en temps de propagation à 25° C, $V_{CC} = 5\ V$, $C_L = 25\ pF$, $R_L = 2k\Omega$. Les circuits Am 25 LS sont spécifiés également :

- de 0° C à 70° C, $V_{CC} = 5\ V \pm 5\ \%$ (4,75 V à 5,25 V), $C_L = 50\ pF$, $R_L = 2k\Omega$.
- de $-$ 55° C à 125° C, $V_{CC} = 5\ V \pm 10\ \%$ (4,5 V à 5,5 V), $C_L = 50\ pF$, $R_L = 2k\Omega$.

Pour la gamme commerciale, AMD donne une variation de 10 % des paramètres de temps de propagation sous l'effet de la température, de 30 % quand on combine les effets de la température et de la variation de tension d'alimentation. Les variations pour la gamme militaire passent respectivement à 30 % et 50 %.

Ceci montre que lors de la conception d'un système, il y a intérêt à ne pas utiliser les composants dans leurs performances maximum.

14. EXEMPLES DE PARAMÈTRES ÉLECTRIQUES ET DYNAMIQUES

Les paramètres électriques et dynamiques de trois types de boîtier ont été extraits du catalogue du constructeur Texas Instruments. Ils concernent le boîtier 74 LS 00 de 4 NAND à 2 entrées, à sortie totem-pôle (fig. 5.24), le boîtier 7407 de 6 "buffers" non inverseurs (amplificateur de charge) à collecteur ouvert et pouvant commander des charges sous une tension élevée, jusqu'à 30 *V* (fig. 5.25) et le boîtier 74 LS 125 de 4 "buffers tri-state", non inverseurs (fig. 5.26).

Les paramètres cités dans ce chapitre sont les plus couramment utilisés. Certains noms peuvent parfois différer d'un constructeur à l'autre. Notons qu'il existe pour les circuits LSI des quantités d'autres paramètres, surtout dynamiques, mais qui n'ont pas de portée générale et se rapportent seulement à un boîtier précis. Le lecteur s'il est déjà familier des paramètres de base étudiés ici pourra faire les adaptations nécessaires et parfois même reconnaître sous un nom ésotérique un paramètre d'usage courant.

Il est important de savoir lire un catalogue de spécifications car on y trouve les caractéristiques maximum d'utilisation, ce qui évite de concevoir des montages anormaux. Ce travail est un peu ingrat mais nécessaire, surtout pour les circuits MSI et LSI.

CARACTÉRISTIQUES	CONDITIONS DE MESURE		TYP MIN	NOM	MAX	UNITE
V_{CC} : Tension d'alimentation			4,75	5	5,25	V
I_{OH} : Courant de sortie, niveau haut					−400	µA
I_{OL} : Courant de sortie, niveau bas					8	mA
T_A : Température de fonctionnement			0		70	°C
V_{IH} : Tension d'entrée, niveau haut			2			V
V_{IL} : Tension d'entrée, niveau bas					0,8	V
V_{IK} : Tension d'écrêtage d'entrée	V_{CC} = Min I_I = − 18 mA				−1,5	V
V_{OH} : Tension de sortie, niveau haut	V_{CC} = Min V_{IL} = V_{IL} max I_{OH} = Max		2,7	3,4		V
V_{OL} : Tension de sortie, niveau bas	V_{CC} = Min	I_{OL} = Max		0,25	0,5	V
	V_{IH} = 2 V	I_{OL} = 4 mA			0,4	
I_I : Courant d'entrée, à la tension d'entrée maximum	V_{CC} = Max V_I = 7 V				0,1	mA
I_{IH} : Courant d'entrée, niveau haut	V_{CC} = Max V_{IH} = 2,7 V				20	µA
I_{IL} : Courant d'entrée, niveau bas	V_{CC} = Max V_{IL} = 0,4 V				−0,4	mA
I_{OS} : Courant de sortie au niveau haut en court-circuit *	V_{CC} = Max		−20		−100	mA
I_{CCH} : Courant total d'alimentation avec sorties au niveau haut	V_{CC} = Max			0,8	1,6	mA
I_{CCL} : Courant total d'alimentation avec sorties au niveau bas	V_{CC} = Max			2,4	4,4	mA
I_{CC} : Courant moyen par porte avec rapport cyclique de 50 %	V_{CC} = Max			0,4		mA
t_{PLH} : Temps de propagation du niveau bas au niveau haut	C_L = 15 pF R_L = 2 kΩ V_{CC} = 5 V T_A = 25 °C			9	15	ns
t_{PHL} : Temps de propagation du niveau haut au niveau bas	C_L = 15 pF R_L = 2 kΩ V_{CC} = 5 V T_A = 25 °C			10	15	ns

* Une sortie à la fois en court-circuit à la masse et pendant une seconde au plus.

Fig. 5.24. — *Paramètres électriques et temporels du boîtier 74 LS 00.*

CARACTÉRISTIQUES	CONDITIONS DE MESURE	TYP MIN	NOM	MAX	UNITE
V_{CC} : Tension d'alimentation		4,75	5	5,25	V
V_{OH} : Tension de sortie, niveau haut				30	V
I_{OL} : Courant de sortie, niveau bas				40	mA
T_A : Température de fonctionnement		0		70	°C
V_{IH} : Tension d'entrée, niveau haut		2			V
V_{IL} : Tension d'entrée, niveau bas				0,8	V
V_{IK} : Tension d'écrêtage	V_{CC} = Min I_I = − 12 mA			−1,5	V
I_{OH} : Courant de sortie, niveau haut	V_{CC} = Min $V_I = V_{IH}$ = 2 V, V_{OH} = Max			250	µA
V_{OL} : Tension de sortie, niveau bas	V_{CC} = Min, I_{OL} = 16 mA			0,4	V
	$V_I = V_{IL}$, I_{OL} = Max			0,7	V
I_I : Courant d'entrée à la tension maximum d'entrée	V_{CC} = Max V_I = 5,5 V			1	mA
I_{IH} : Courant d'entrée, niveau haut	V_{CC} = Max V_{IH} = 2,4 V			40	µA
I_{IL} : Courant d'entrée, niveau bas	V_{CC} = Max V_{IL} = 0,4 V			−1,6	mA
I_{CCH} : Courant total d'alimentation avec sorties au niveau haut			29	41	mA
I_{CCL} : Courant total d'alimentation avec sorties au niveau bas			21	30	mA
I_{CC} : Courant moyen par porte avec rapport cyclique de 50 %				4,17	mA
t_{PLH} : Temps de propagation du niveau bas au niveau haut	C_L = 15 pF R_L = 110 Ω Tension de charge : V_{CC} V_{CC} = 5 V T_A = 25 °C		6	10	ns
t_{PHL} : Temps de propagation du niveau haut au niveau bas	Identique à t_{PLH}		20	30	ns

Fig. 5.25. — *Paramètres électriques et temporels du boîtier de buffers à collecteur ouvert 7407.*

CARACTÉRISTIQUES	CONDITIONS DE MESURE		TYP MIN	NOM	MAX	UNITE
V_{CC} : Tension d'alimentation			4,75	5	5,25	V
I_{OH} : Courant de sortie, niveau haut					−2,6	mA
I_{OL} : Courant de sortie, niveau bas					24	mA
T_A : Température de fonctionnement			0		70	°C
V_{IH} : Tension d'entrée, niveau haut			2			V
V_{IL} : Tension d'entrée, niveau bas					0,8	V
V_{IK} : Tension d'entrée d'écrêtage	V_{CC} = Min I_I = − 18 mA				−1,5	V
V_{OH} : Tension de sortie, niveau haut	V_{CC} = Min V_{IH} = 2 V V_{IL} = V_{IL} max I_{OH} = Max		2,4			V
V_{OL} : Tension de sortie, niveau bas	V_{CC} = Min V_{IH} = 2 V V_{IL} = V_{IL} max	I_{OL} = Max		0,35	0,5	V
		I_{OL}=12 mA		0,25	0,4	V
I_{OZ} : Courant de sortie à l'état haute impédance	V_{CC} = Max V_{IL} = V_{IL} max V_{IH} = 2V	V_O = 2,4 V			20	µA
		V_O = 0.4 V			−20	µA
I_I : Courant d'entrée à la tension maximum d'entrée	V_{CC} = Max V_I = 7 V				0,1	mA
I_{IH} : Courant d'entrée, niveau haut	V_{CC} = Max V_{IH} = 2,7 V				20	µA
I_{IL} : Courant d'entrée, niveau bas	V_{CC} = Max V_{IL} = 0,4 V				−0,4	mA
I_{OS} : Courant de sortie au niveau haut en court-circuit *	V_{CC} = Max		−40		−225	mA
I_{CC} : Courant total d'alimentation	V_{CC} = Max donnée = OV control = 4,5 V			11	20	mA
t_{PLH} : Temps de propagation du niveau bas au niveau haut	C_L = 45 pF R_L = 667 Ω			9	15	ns
t_{PHL} : Temps de propagation du niveau haut au niveau bas				7	18	ns
t_{PZH} : Temps de validation de sortie à l'état haut				12	20	ns
t_{PZL} : Temps de validation de sortie à l'état bas				15	25	ns
t_{PHZ} : Temps de désactivation de l'état haut	C_L = 5 pF R_L = 667 Ω				20	ns
t_{PLZ} : Temps de désactivation de l'état bas					20	ns

* Une sortie à la fois en court-circuit à la masse et pendant une seconde au plus.

Fig. 5.26. — *Paramètres électriques et temporels du boîtier tri-state 74 LS 125.*

Chapitre 6

EXEMPLES D'UTILISATION DES PARAMÈTRES ÉLECTRIQUES

1. CONNEXION DE PORTES EN "ET" CABLÉ A COLLECTEUR OUVERT

Plusieurs portes (m) à collecteur ouvert sont réunies par leurs sorties et commandent plusieurs entrées (p) de portes. Le problème consiste à déterminer les valeurs de m et p, et de la résistance commune R_L à mettre comme charge (fig. 6.1) en fonction des paramètres de tension et de courant donnés par les constructeurs dans leurs catalogues.

1.1. Relation entre m, p, R_L pour le niveau haut

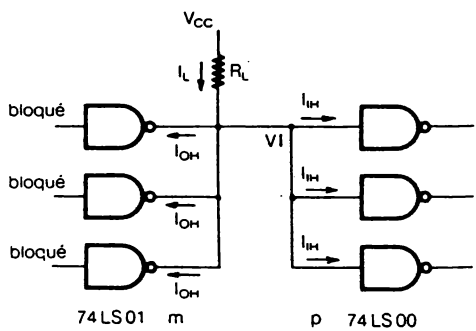


Fig. 6.1. — *Configuration avec les portes bloquées.*

On écrit :

— que le courant I_L fourni par la résistance R_L est égal aux courants I_{OH} consommés par les sorties bloquées (rappelons qu'une porte à sortie totem pôle fournit du courant alors qu'une porte à collecteur ouvert absorbe du courant au niveau V_{OH}) et aux courants d'entrée I_{IH}.

— Que le niveau de tension V_I du point commun est comparable au niveau de sortie d'une porte à l'état haut soit V_{OH}.

$$I_L = mI_{OH} + pI_{IH} \qquad (m \text{ sorties, } p \text{ entrées})$$

$$V_I \geqslant V_{OH\ min}$$

$$V_I = V_{CC} - R_L I_L$$

$$V_I = V_{CC} - (R_L.m.I_{OH} + R_L.p.I_{IH})$$

donc :

$$V_{CC} - R_L (m I_{OH} + p I_{IH}) \geqslant V_{OH\ min}$$

D'où :

$$R_L \leqslant \frac{V_{CC} - V_{OH\ min}}{m I_{OH} + p I_{IH}} = R\ max$$

1.2. Relation entre m, p, R_L pour le niveau bas

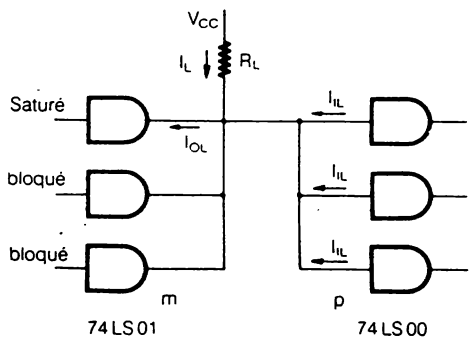


Fig. 6.2. — *Configuration pour le niveau bas.*

Pour que le montage fonctionne correctement, il faut :

- que la somme des courants I_L et des courants d'entrée I_{IL} soit inférieure ou égale au courant maximum I_{OL} que peut absorber une sortie,
- que le niveau d'entrée V_I soit inférieur ou égal à celui d'une sortie au niveau bas soit $V_{OL\ max}$.

Cela se traduit par :

$$I_{OL\ max} \geqslant I_L + p\, I_{IL}$$

$$V_{CC} - R_L\, I_L \leqslant V_{OL\ max}$$

$$V_{CC} - R_L\,(I_{OL\ max} - p\, I_{IL}) \leqslant V_{OL\ max}\ ,$$

$$R_L \geqslant \frac{V_{CC} - V_{OL\ max}}{I_{OL\ max} - p\, I_{IL}} = R_{min}$$

En conclusion, R_L doit être compris entre deux valeurs R_{min} et R_{max} :

$$R_{min} \leqslant R_L \leqslant R_{max}$$

1.3. Commande de plusieurs charges

La relation $\dfrac{V_{CC} - V_{OL\ max}}{I_{OL\ max} - p\, I_{IL}} \leqslant R_L \leqslant \dfrac{V_{CC} - V_{OH\ min}}{m\, I_{OH} + p\, I_{IH}}$

permet de calculer les valeurs R_{min} et R_{max} qui encadrent R_L. Prenons un montage avec huit sorties collecteur ouvert 74 LS 01 alimentant 10 entrées 74 LS 00. A un instant donné, il n'y a que deux situations : les huit sorties sont bloquées, ou sept sorties sont bloquées et une est saturée. La figure 6.3 donne la valeurs des paramètres des portes utilisées.

Paramètres / Circuits	V_{CC}	$V_{OH\,min}$	$V_{OL\,max}$	I_{OH}	I_{IH}	I_{OL}	I_{IL}
74 LS 00	5 V	2,4 V	0,5 V	400 µA	20 µA	—	0,4 mA
74 LS 01	5 V	—	—	250 µA	—	8 mA	—

Fig. 6.3. – *Paramètres électriques des circuits 74 LS 00 et 74 LS 01.*

A l'état haut, la résistance maximum est déterminée par:

$$R_{max} = \frac{V_{CC} - V_{OH\ min}}{m\, I_{OH} + p\, I_{IH}}$$

$$R_{max} = \frac{5 - 2{,}4}{8 \times 0{,}250 + 10 \times 0{,}020} = 1{,}18\ k\Omega$$

A l'état bas, la résistance minimum vaut:

$$R_{min} = \frac{V_{CC} - V_{OL}}{I_{OL\ max} - p\, I_{iL}}$$

$$R_{min} = \frac{5 - 0{,}5}{8 - 10 \times 0{,}4} = 1{,}12\ k\Omega$$

On obtient donc:

$$1{,}12\ k\Omega \leqslant R_L \leqslant 1{,}18\ k\Omega$$

Si l'on se reporte au tableau de la figure 6.4 qui donne la gamme des valeurs normalisées des résistances à 10 %, il n'est pas possible de trouver une valeur normalisée encadrée entre les deux valeurs précédemment calculées. Il en serait de même dans les gammes à 5 % ou 2 % dont les coûts deviennent d'ailleurs très rapidement prohibitifs. Le montage comprenant 8 sorties à collecteur ouvert pour commander 10 charges ne peut donc fonctionner.

10 10^n	33 10^n
12 10^n	39 10^n
15 10^n	47 10^n
18 10^n	56 10^n
22 10^n	68 10^n
27 10^n	82 10^n

pour n = 0 (10 Ω, 12 Ω, 15 Ω)

pour n = 3 (10 kΩ, 12 kΩ, 15 kΩ)

Fig. 6.4. — *Valeurs normalisées des résistances à 10 % (série E 12).*

Reprenons le même problème avec dix charges commandées, mais seulement quatre sorties collecteurs ouverts réunies. Nous avons toujours:

$$R_{min} = \frac{5 - 0{,}5}{8 - 10 \times 0{,}4} = 1{,}12\ k\Omega$$

Mais :

$$R_{max} = \frac{5 - 2{,}4}{4 \times 0{,}25 + 10 \times 0{,}02} = 2{,}16\ K\Omega$$

Soit :

$1{,}12\ K\Omega \leqslant R_L \leqslant 2{,}16\ K\Omega$

La résistance moyenne R_L doit être à un même pourcentage x au-dessus de R_{min} et au-dessous de R_{max} soit $R_L = R_{min}(1 + x)$ et $R_L = \frac{R_{max}}{(1 + x)}$ d'où $R_L = \sqrt{R_{min} \times R_{max}}$.

Nous prendront alors pour R_L la valeur de résistance dans la gamme à 10 % la plus proche de la moyenne géométrique $\sqrt{1{,}12 \times 2{,}16} = 1{,}55$ soit 1,5 K.

1.4. Commande d'une seule charge

Lorsque l'on réalise un ET câblé, c'est généralement pour économiser des portes de réunion. On cherche donc à en connecter un maximum ensemble. La seule contrainte consiste à pouvoir commander au moins une porte. S'il y a plusieurs portes à commander, on utilise un amplificateur de charge (buffer) ou encore, on dédouble la porte commandée (fig. 6.5).

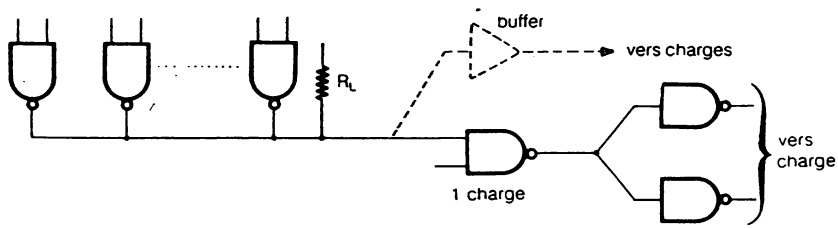


Fig. 6.5. — *Configuration avec le maximum de portes en ET câblé.*

Supposons donc que nous ne commandions qu'une seule charge, et déterminons la valeur optimum de R_L lorsque nous avons de 2 à m portes collecteurs ouverts réunies entre elles ? Quel est le nombre de portes m maximum que nous pouvons réunir ensemble ? Tout d'abord, de façon à se

ménager une certaine sécurité de fonctionnement, compte tenu de la précision des résistances, des variations de la source d'alimentation et des tolérances sur les différents paramètres électriques de courants et tension nous nous imposons une valeur de R_L prise à 20 % au-dessus de R_{min} et à 20 % au-dessous de R_{max}.

Nous avons $R_{min} = \dfrac{V_{CC} - V_{OL\ max}}{I_{OL\ max} - I_{IL}}$

$$R_{max} = \frac{V_{CC} - V_{OH\ min}}{m\, I_{OH} + I_{IH}}$$

Ces deux relations nous permettent d'établir le tableau de la figure 6.6 pour m compris entre 2 et 11. La valeur moyenne de R_L s'obtient par la moyenne géométrique de R_{min} et R_{max} et l'on choisit dans le tableau de la figure 6.4 la valeur normalisée la plus proche de cette moyenne à condition qu'elle soit comprise dans la fourchette R_{min} + 20 %, R_{max} − 20 %.

m	R_{min_Ω}	R_{max_Ω}	R_{min} + 20 %	R_{max} − 20 %	Moyenne	R_L
2	592	5000	710	4000	1685	1800
3	592	3376	710	2701	1384	1500
4	592	2549	710	2039	1203	1200
5	592	2047	710	1637	1078	1000
6	592	1710	710	1368	985	1000
7	592	1468	710	1175	913	1000
8	592	1287	710	1029	854	820
9	592	1145	710	916	806	820
10	592	1031	710	825	765	820
11	592	938	710	750	729	pas de solution

Fig. 6.6. — *Calcul de la valeur optimum de R_L pour différentes valeurs de m.*

Le graphique de la figure 6.7 illustre le tableau de la figure 6.6 en mettant en évidence le rétrécissement de la plage disponible pour effectuer le choix de R_L avec les valeurs croissantes de m.

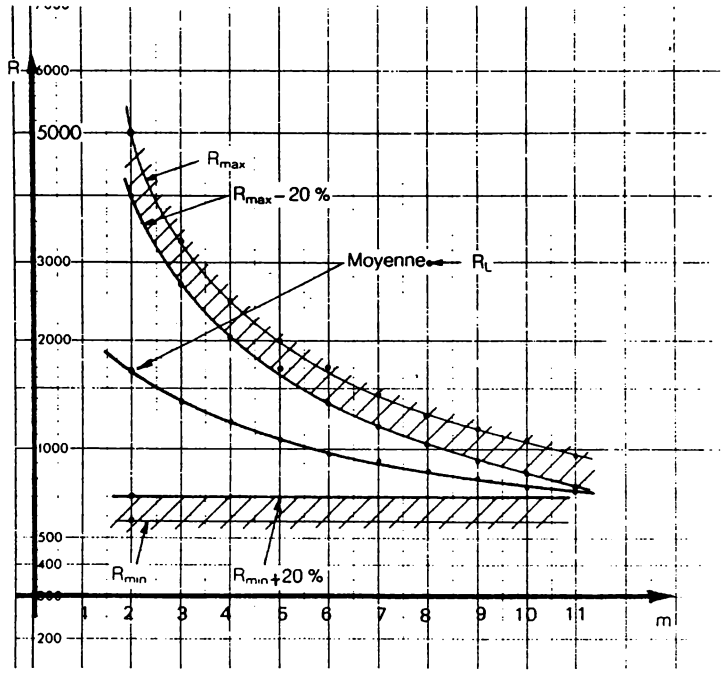


Fig. 6.7 – *Choix de la valeur de R_L en fonction de m.*

2. CONNEXION DE PORTES "TRI-STATE" EN COMMUN

Établissons un parallèle entre les montages à collecteur ouvert et les montages à portes tri-state.

2.1. Calcul au niveau haut

On suppose qu'une seule sortie est active, les autres étant bloquées. Les portes tri-state bloquées auxquelles on applique par l'intermédiaire d'une autre porte tri-state active un niveau haut V_{OH}, se comportent comme une entrée de porte au niveau haut et absorbent un courant I_{OZH}. Donc $I_{OH} \geqslant (m-1)\, I_{OZH} + p\, I_{IH}$:

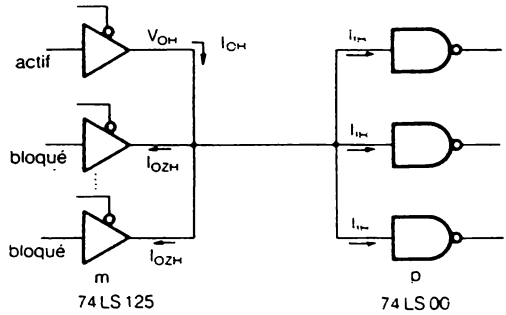


Fig. 6.8. — *Circulation des courants pour des portes tri-state au niveau haut.*

2.2. Calcul au niveau bas

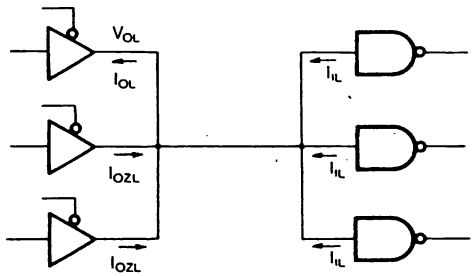


Fig. 6.9. — *Circulation des courants pour des portes tri-state au niveau bas.*

Une porte tri-state bloquée, à laquelle on applique, par l'intermédiaire d'une autre porte tri-state active un niveau bas, se comporte comme une entrée de porte au niveau bas et fournit un courant I_{OZL} comme une entrée au niveau bas. Donc :

$$I_{OL} \geqslant (m-1)\ I_{OZL} + p\ I_{IL}$$

2.3. Commande de plusieurs charges

Reprenons le même exemple qu'avec les collecteurs ouverts $m = 8$, $p = 10$.

	74 LS 125	74 LS 00
I_{OH}	−2,6 mA	− 400 μA
I_{OZH}	20 μA	—
I_{IH}	20 μA	20 μA
I_{OL}	24 mA	8 mA
I_{OZL}	20 μA	—
I_{IL}	−0,4 mA	−0,4 mA

Fig. 6.10. — *Quelques paramètres des tri-state 74 LS 125 et porte 74 LS 00.*

Vérifions si les inégalités précédentes sont satisfaites pour les paramètres spécifiés au tableau de la figure 6.10 :

$I_{OH} \geqslant (m-1)\, I_{OZH} + p\, I_{IH}$ soit

$2{,}6\ mA \geqslant 7 \times 0{,}020 + 10 \times 0{,}020$

$2{,}6\ mA \geqslant 0{,}34\ mA$ largement satisfait

$I_{OL} \geqslant (m-1)\, I_{OZL} + p\, I_{IL}$

$24\ mA \geqslant 7 \times 0{,}020 + 10 \times 0{,}4$

$24\ mA \geqslant 4{,}14\ mA$ largement satisfait.

En conclusion, la porte tri-state permet de commander sans problème un grand nombre de charges.

2.4. Commande d'une seule charge

Supposons qu'il n'y a qu'une seule charge à commander et calculons le nombre de sorties que l'on peut connecter ensemble. Les formules deviennent :

$$I_{OH} = (m-1)\, I_{OZH} + I_{IH}$$

$$I_{OL} = (m-1)\, I_{OZL} + I_{IL}$$

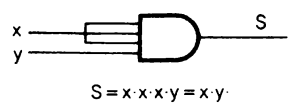
$$m - 1 = \frac{I_{OH} - I_{IH}}{I_{OZH}} \simeq \frac{I_{OH}}{I_{OZH}} = \frac{2{,}6}{0{,}02} = 130$$

$$m - 1 = \frac{I_{OL} - I_{IL}}{I_{OZL}} \simeq \frac{I_{OL}}{I_{OZL}} = \frac{24}{0{,}02} = 1200$$

La deuxième valeur 1200 est à éliminer puisque la première limite à 130 charges. Même avec 130 charges, on obtient une bien meilleure performance qu'avec les portes à collecteur ouvert (11 charges). Ce nombre de 130 peut paraître excessif. En fait, dans la pratique, on obtient couramment des groupements de plusieurs dizaines de portes dans les montages répétitifs comme les cartes de mémoire. Cependant, la réunion de plusieurs portes tri-state accroît la capacité de sortie de l'ensemble, ce qui augmente les temps de transitions des signaux d'un niveau à un autre. Aussi les constructeurs spécifient-ils les retards des circuits pour des charges capacitives données, notamment pour les boîtiers mémoires qui possèdent des sorties tri-state.

3. POLARISATION DES ENTRÉES

En logique TTL, on ne doit pas laisser les entrées inutilisées en l'air car ces entrées risquent de se comporter comme des antennes et amener du bruit. Ainsi, des entrées de AND ou NAND non utilisées seront polarisées avec des 1, des entrées de OR avec des 0, et d'une manière générale, les entrées inutilisées d'un circuit quelconque seront placées dans leur état inactif (Preset ou Clear de bascule à l'état 1 par exemple). Pour polariser les entrées, il existe plusieurs solutions. Pour des AND ou NAND, on peut réunir les entrées utilisées à des entrées inutilisées (fig. 6.11). Du point de vue électrique au niveau haut, les entrées inutilisées absorbent un courant I_{IH} qui s'ajoute au courant I_{IH} de l'entrée utilisée avec qui elle est connectée (fig. 6.12a). Par contre, au niveau bas, le courant I_{IL}, qui provient de la résistance de base du transistor à émetteur multiple se subdivise entre les différents émetteurs, la sortie n'a pas à absorber de courant supplémentaire (fig. 6.12b). Il n'y a donc pas de charge supplémentaire au niveau bas. Cependant comme on ne voit pas bien à quoi serviront une porte AND ou NAND dont une entrée serait constamment au niveau bas, on est obligé de compter comme charges supplémentaires les entrées inutilisées connectées aux entrées utilisées.



$S = x \cdot x \cdot x \cdot y = x \cdot y$

Fig. 6.11. — *Polarisation d'entrées non utilisées d'une porte NAND.*

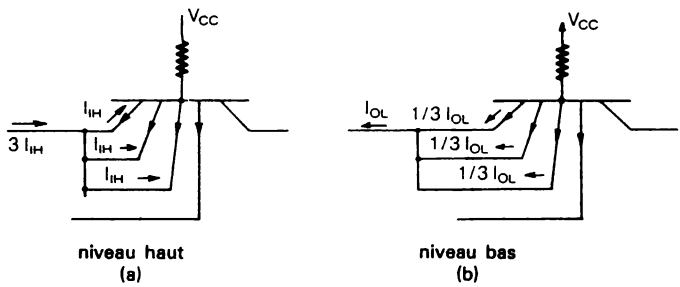


Fig. 6.12. — *Polarisation d'entrées non utilisées d'une porte AND. Circulation des courants.*

Une autre solution est de mettre une résistance de rappel (pull-up) au + 5 *V* pour une entrée inactive au niveau 1. Texas préconise une résistance de 1 *kΩ*, alimentant jusqu'à 25 entrées inutilisées. Le but est de définir un niveau compris entre V_{OH} et 4,5 *V*. Pour une porte standard 74 00, le courant inverse d'entrée augmente très vite avec la tension d'alimentation (fig. 6.13).

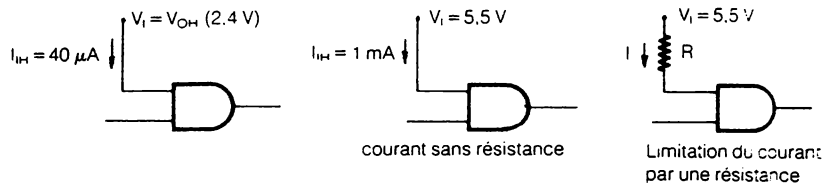


Fig. 6.13. — *La résistance limite le courant.*

Par contre, pour une porte LS, le courant augmente beaucoup moins : il passe de 20 *μA* pour $V_I = V_{OH}$ à 100 *μA* pour $V_I = 7$ *V*. On peut se dispenser d'une résistance de rappel et mettre les entrées inutilisées au + 5 *V* directement.

Une autre solution, bien plus commode que la résistance commune est de prendre une sortie de porte et la maintenir à 0 ou 1, avec une commande correcte des entrées. Le 1 ou le 0 permanent ainsi obtenu sert à polariser les

entrées inutilisées, en se contentant d'appliquer les règles de charge habituelles. Classiquement, on utilise un inverseur (fig. 6.14).

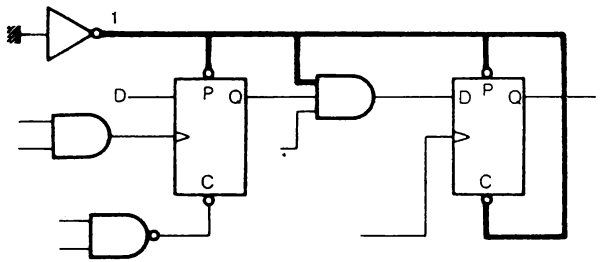


Fig. 6.14. — *Polarisation d'entrées inutilisées avec un inverseur.*

Chapitre 7

FONCTIONS COMBINATOIRES USUELLES ET CIRCUITS ASSOCIÉS

1. INTRODUCTION

La pratique montre que dans les schémas combinatoires complexes, on peut distinguer deux types de fonctions :

- Des fonctions caractéristiques du problème à résoudre et que l'on réalise habituellement avec des opérateurs élémentaires ET, OU, NOR, NAND, inverseurs.
- Des fonctions communes à de nombreux problèmes combinatoires, réalisables bien sûr avec les opérateurs élémentaires, mais que les constructeurs, en raison même de leur fréquence d'emploi, ont intégré dans des boîtiers.

Ces fonctions communes portent sur plusieurs éléments binaires à la fois. Il s'agit de fonctions comme la comparaison de deux quantités (égalité, supériorité, infériorité), d'analyse d'information (codage, décodage), d'aiguillage d'information, de combinaison d'information selon des opérations logiques ou arithmétiques.

Nous examinons ici les opérateurs élémentaires, puis les fonctions plus complexes couramment utilisées. Enfin, une annexe au chapitre 7 donne un

aperçu assez complet des circuits intégrés commercialisés en technologie TTL-LS.

2. LES OPÉRATEURS ÉLÉMENTAIRES

2.1. Les types d'opérateurs

On trouve en boîtiers intégrés, des inverseurs, des amplificateurs séparateurs, des portes AND, des portes OR, des portes NAND, des portes NOR, ainsi que des sommes modulo 2 (OU exclusif).

Le nombre d'opérateurs par boîtier dépend du nombre d'entrées et de sorties de chaque opérateur et du nombre de broches disponibles dans le boîtier. Par exemple, dans un boîtier à 14 broches, on a :

- deux broches pour l'alimentation et la masse,
- 12 broches permettant la connexion de 4 portes NAND à 2 entrées et 1 sortie (4 × 3), 6 inverseurs (6 × 2), 1 porte NAND à 8 entrées (8 + 1) etc...

L'annexe de ce chapitre montre qu'en technologie TTL-LS, il n'existe que quelques portes AND, NOR, une porte OR mais une grande variété de porte NAND de 2 à 13 entrées.

2.2. Caractéristiques électriques des opérateurs

On peut distinguer au niveau des opérateurs :

- Des opérateurs à sortance et entrance normale
- Des opérateurs collecteur ouvert à tension de sortie 5 *V*, 15 *V* ou 30 *V* capables de débiter des courants plus ou moins importants (amplificateurs).
- Des opérateurs à commande 3 états.
- Des opérateurs avec entrée trigger de Schmitt.

L'annexe du chapitre 7 met en évidence la diversité des opérateurs dont peut disposer le concepteur.

3. COMPARAISON ENTRE INFORMATIONS

3.1. Comparaison à zéro ou décodage à zéro

Soit une quantité binaire A représentée par n éléments binaires a_0, a_1, a_2 ... a_{n-1}. Pour déterminer si l'information A est nulle, il faut calculer une fonction f telle que :

$f = 1$ si $A = 0$ et $f = 0$ si $A \neq 0$

A est nul si tous les eb sont nuls.

$A = 0$ si $a_0 = 0$, et $a_1 = 0$, et $a_2 = 0$, ... et $a_{n-1} = 0$

La fonction f s'écrit alors :

$$f = \bar{a}_0 . \bar{a}_1 . \bar{a}_2 \ \ldots \ \overline{a_{n-1}}$$

Nous voyons que si un seul eb ou plusieurs sont en 1, la fonction vaut 0. f est appelée décodage à 0 de A.

La fonction g, inverse de f, telle que $g = 1$ si $A \neq 0$ et $g = 0$ si $A = 0$ est telle que :

$$g = \overline{\bar{a}_0 . \bar{a}_1 . \bar{a}_2 \ \ldots \ \overline{a_{n-1}}} = a_0 + a_1 + a_2 + \ \ldots \ a_{n-1}$$

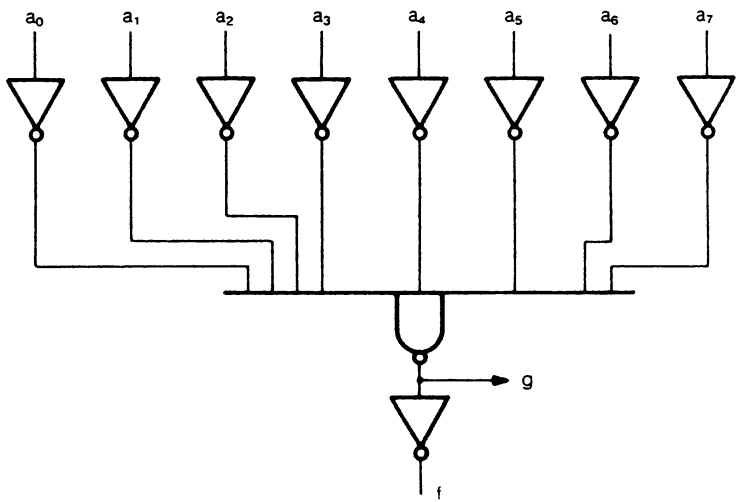


Fig. 7.1. — *Exemple de décodage à 0.*

Les fonctions f et g, qui n'existent pas sous forme d'un boîtier, se réalisent avec des portes élémentaires (fig. 7.1) ou avec des circuits d'égalité (voir § 3.2).

3.2. Égalité entre deux informations

3.2.1. Principe

Nous avons défini dans le chapitre 1 la fonction égalité de deux variables x et y rappelée à la figure 7.2.

$$f(x,y) = x \odot y = \overline{x \oplus y}$$

Cette fonction s'étend facilement à plus de deux éléments binaires. Deux quantités A et B à n éléments binaires chacune sont égales si tous les éléments binaires homologues sont égaux.

$f = 1$ si $(a_0 = b_0)$ ET $(a_1 = b_1)$ ET $(a_2 = b_2)$ ...

$$f = \overline{a_0 \oplus b_0} \cdot \overline{a_1 \oplus b_1} \cdot \overline{a_2 \oplus b_2} \cdot \ldots \cdot \overline{a_{n-1} \oplus b_{n-1}}$$

a_i	b_i	égalité
0	0	1
0	1	0
1	0	0
1	1	1

Fig. 7.2. — *Fonction égalité de deux variables.*

3.2.2. Exemple

Le constructeur de circuits AMD propose un circuit intégré à 20 broches qui réalise l'égalité sur huit éléments binaires (fig. 7.3). En fait, le circuit calcule $\overline{f}$ (sortie E_{out}). Il permet l'extension au-delà de 8 eb par l'entrée E_{in}. On a :

$$E_{out} = \overline{\overline{E}_{in} (\overline{A}_0 \oplus \overline{B}_0)(\overline{A}_1 \oplus \overline{B}_1) \ldots (\overline{A}_7 \oplus \overline{B}_7)}$$

A l'inversion près et au produit par E_{in} qui permet la mise en cascade des boîtiers, on retrouve la formule générale.

Remarquons que ce circuit permet le décodage à 0 d'un nombre A. Il suffit alors de mettre la valeur A sur les entrées A et 0 sur les entrées B.

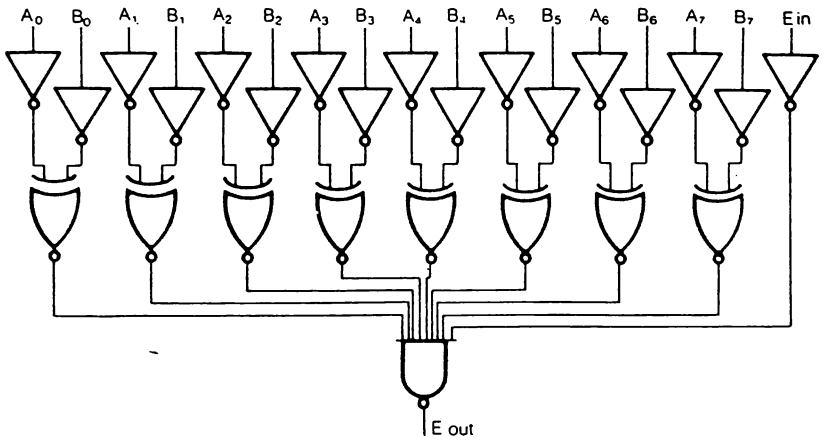


Fig. 7.3. — *Circuit de comparaison Am 25 LS 2521.*

3.3. Comparaison arithmétique de deux nombres *A* et *B*

3.3.1. Principe pour 4 eb

Il s'agit de classer deux nombres A et B exprimés en binaire pur et de calculer les fonctions $A = B$, $A > B$ ou $A < B$. Ceci peut se réaliser à l'aide d'un circuit combinatoire qui compare chacun des eb pris deux à deux. Prenons deux nombres A et B de 4 eb :

$$A = a_3\ a_2\ a_1\ a_0 \qquad B = b_3\ b_2\ b_1\ b_0$$

A est plus grand que B si $a_3 > b_3$ ou si $a_3 = b_3$, $a_2 > b_2$ ou si $a_3 = b_3$, $a_2 = b_2$, $a_1 > b_1$ ou si $a_3 = b_3$, $a_2 = b_2$, $a_1 = b_1$, $a_0 > b_0$.

D'où la forme de f :

$$f = (a_3 > b_3) + (a_3 = b_3).(a_2 > b_2) + (a_3 = b_3).(a_2 = b_2).(a_1 > b_1) + (a_3 = b_3).(a_2 = b_2).(a_1 = b_1).(a_0 > b_0)$$

La figure 7.4 donne le tableau de la fonction $a_i > b_i$, qui permet d'exprimer f.

a_i	b_i	$a_i > b_i$
0	0	0
0	1	0
1	0	1
1	1	0

$a_i > b_i = a_i \bar{b}_i$

Fig. 7.4. — *Tableau de vérité de la fonction supériorité.*

$$f = a_3 \bar{b}_3 + \overline{(a_3 \oplus b_3)}\, a_2 \bar{b}_2 + \overline{(a_3 \oplus b_3)}\, \overline{(a_2 \oplus b_2)} \cdot a_1 \bar{b}_1 + \overline{(a_3 \oplus b_3)} \cdot \overline{(a_2 \oplus b_2)} \cdot \overline{(a_1 \oplus b_1)}\, a_0 \bar{b}_0$$

Un raisonnement semblable s'applique pour les cas de $A < B$. Le cas $A = B$ a déjà été étudié au paragraphe 3.2.

3.3.2. Extension au-delà de 4 eb

L'analyse précédente s'étend au-delà de 4 eb. On découpe alors les nombres en tranches de 4 eb. Prenons deux nombres A et B à 8 eb.

$$A = a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$$

$$B = b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$$

notés d'une manière condensée A_0^7 et B_0^7, a_7 et b_7 étant les poids forts.

Si $A_4^7 > B_4^7$, alors $A > B$ quels que soient A_0^3 et B_0^3.

Si $A_4^7 < B_4^7$, alors $A < B$ quels que soient A_0^3 et B_0^3

Si $A_4^7 = B_4^7$, et $A_0^3 > B_0^3$, alors $A > B$

Si $A_4^7 = B_4^7$, et $A_0^3 < B_0^3$, alors $A < B$

Si $A_4^7 = B_4^7$, et $A_0^3 = B_0^3$, alors $A = B$

On voit que lorsque la tranche de poids fort permet de prendre la décision, on n'a pas besoin de A_0^3 et B_0^3. Par contre, quand les poids forts sont égaux, la décision globale dépend de la comparaison de la tranche de poids faible.

$$(A > B) = (A_4^7 > B_4^7) + (A_4^7 = B_4^7).(A_0^3 > B_0^3)$$

$$(A < B) = (A_4^7 < B_4^7) + (A_4^7 = B_4^7).(A_0^3 < B_0^3)$$

$$(A = B) = (A_4^7 = B_4^7).(A_0^3 = B_0^3)$$

Un circuit de comparaison cascadable doit donc avoir les entrées d'information A et B et des entrées amenant la comparaison de la tranche inférieure.

3.3.3. Exemple

Les figures 7.5 et 7.6 présentent le schéma bloc et le tableau de vérité d'un circuit intégré répondant à l'analyse précédente: le 74 LS 85. Les entrées de cascadage permettent la comparaison de nombres de longueur quelconque par association de boîtiers entre eux. On trouvera en annexe de ce chapitre, d'autres comparateurs intégrés répondant au même principe de calcul.

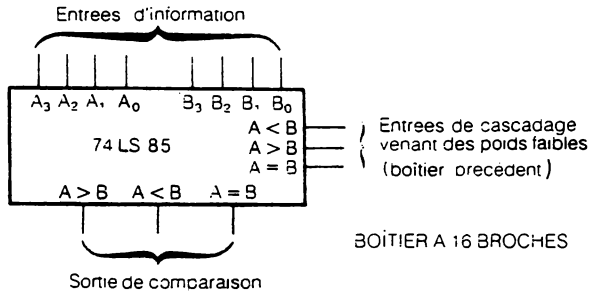


Fig. 7.5. — *Schéma symbolique du comparateur 74 LS 85.*

Nombres à comparer				Entrées de cascadage			Sorties		
a_3, b_3	a_2, b_2	a_1, b_1	a_0, b_0	A>B	A<B	A=B	A>B	A<B	A=B
$a_3 > b_3$	X	X	X	X	X	X	1	0	0
$a_3 < b_3$	X	X	X	X	X	X	0	1	0
$a_3 = b_3$	$a_2 > b_2$	X	X	X	X	X	1	0	0
$a_3 = b_3$	$a_2 < b_2$	X	X	X	X	X	0	1	0
$a_3 = b_3$	$a_2 = b_2$	$a_1 > b_1$	X	X	X	X	1	0	0
$a_3 = b_3$	$a_2 = b_2$	$a_1 < b_1$	X	X	X	X	0	1	0
$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 > b_0$	X	X	X	1	0	0
$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 < b_0$	X	X	X	0	1	0
$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 = b_0$	1	0	0	1	0	0
$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 = b_0$	0	1	0	0	1	0
$a_3 = b_3$	$a_2 = b_2$	$a_1 = b_1$	$a_0 = b_0$	0	0	1	0	0	1

Fig. 7.6. — *Table de vérité du comparateur 74 LS 85.*

4. TRANSFORMATION DE CODES

4.1. Notions

Nous avons vu aux chapitres 2 et 3 que l'on peut représenter l'information codée de différentes façons. Lorsque l'on traite cette information, on a fréquemment besoin de passer d'un code à un autre. Pour les codes les plus courants, les constructeurs ont conçu des circuits intégrés. On trouve ainsi :

- Des circuits qui "décodent" une combinaison codée en binaire pur, en décimal codé binaire, en code de Gray en une combinaison 1 parmi N. Une seule sortie du circuit est active à la fois, celle dont le rang correspond à la valeur codée présentée. Ces circuits portent le nom de "décodeur".
- Des circuits qui font l'opération inverse. Un code 1 parmi N (1 entrée active parmi N) est transformée en un code binaire pur dont la valeur correspond au rang de l'entrée active. Si plusieurs entrées sont actives, on leur affecte une priorité et le circuit réalise l'opération seulement sur la plus prioritaire. Ce circuit effectue l'opération d'encodage (contraire du décodeur). On l'appelle "encodeur".
- Des circuits qui transforment un code décimal codé binaire en code n parmi 7 pour représenter visuellement les chiffres de 0 à 9 avec des afficheurs lumineux à sept segments.
- Des circuits permettant le passage du code décimal au code binaire et inversement. On les appelle "convertisseurs de code".

Examinons plus en détail ces différentes fonctions.

4.2. Fonction de décodage binaire

4.2.1. Principe

Une quantité codée en binaire avec n eb peut représenter $N = 2^n$ valeurs. La fonction de décodage consiste à transformer un codage binaire pur en codage 1 parmi N (avec $N = 2^n$). Le code 1 parmi N est appelé aussi mode positionné. Le circuit a n entrées, 2^n sorties, et éventuellement une ou plusieurs entrées G validant les fonctions de sortie.

Le tableau de vérité de la figure 7.7 et le schéma symbolique de la figure 7.8 appliqués à $n = 3$, résument les propriétés des décodeurs binaires.

Validation	Codage binaire (entrées)			Code 1 parmi N (Sorties)							
G	2^2	2^1	2^0	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0
	C	B	A								
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Fig. 7.7. — *Décodage binaire, table de vérité.*

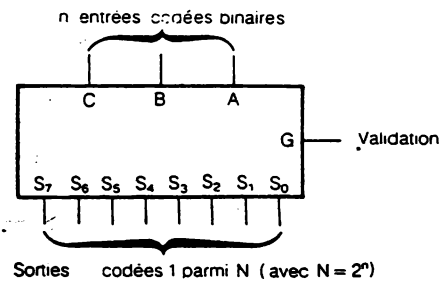


Fig. 7.8. — *Décodeur binaire, schéma symbolique.*

Si $G = 0$, toutes les sorties du décodeur sont inactives quelque soit l'état de l'information binaire sur ABC.

Si $G = 1$, pour une valeur binaire i sur les entrées A, B, C, la sortie de rang i est active et toutes les autres sont inactives. L'équation d'une sortie de rang i s'écrit: $S_i = G.\ (... \ CBA = i)$.

4.2.2. Exemple de décodeur

Le décodeur 74 LS 138 répond aux principes généraux précédents. Les figures 7.9, 7.10 et 7.11 en donnent respectivement le schéma symbolique, le schéma interne et le tableau de vérité.

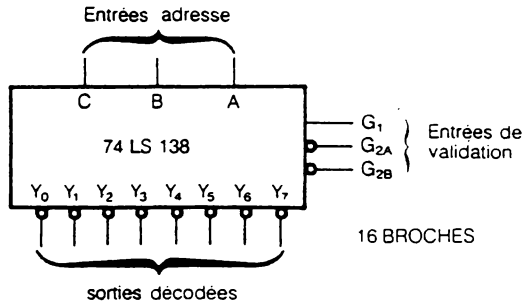


Fig. 7.9. — *Schéma symbolique du décodeur 74 LS 138.*

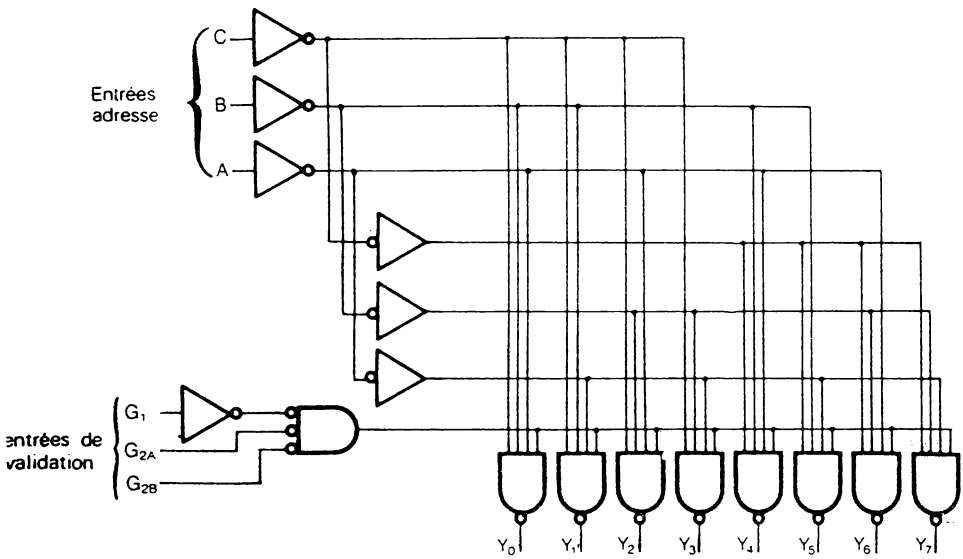


Fig. 7.10. — *Schéma interne du décodeur 74 LS 138.*

Entrées					Sorties							
validation		Adresse										
G_1	G_2*	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

* $G_2 = G_{2A} + G_{2B}$

Fig. 7.11. — *Tableau de vérité du décodeur 74 LS 138.*

Les entrées C, B, A, appelées entrées de sélection, ou encore entrées d'adresse, ont respectivement un poids de 2^2, 2^1, 2^0. Aux trois entrées A, B, C, correspondent donc $2^3 = 8$ sorties notées Y_0 à Y_7. Les entrées G_1, G_{2A}, G_{2B} servent à valider les sorties. (Elles sont validées par $G_1\ \overline{G_{2A}}\ \overline{G_{2B}}$.) Notons que l'on rencontre au niveau des portes traitant les entrées G_1, G_{2A} et $\overline{G_{2B}}$, les symboles logiques dérivés des symboles fondamentaux: une porte ET avec des petits ronds aux entrées. Elle réalise le ET des fonctions inverses. Les sorties du décodeur, issues de portes NAND sont à 0 quand on les sélectionne et à 1 dans le cas contraire.

Calculons à titre d'exercice la sortie Y_3:

$$Y_3 = \overline{\overline{C}\ B\ A\ \overline{G_{2A}}\ \overline{G_{2B}}\ G_1} \text{ soit}$$

$$Y_3 = \overline{G_1\ \overline{G_{2A}}\ \overline{G_{2B}}\ (C\ B\ A = 3)}$$

D'une manière générale:

$$Y_i = \overline{G_1\ \overline{G_{2A}}\ \overline{G_{2B}}\ (C\ B\ A = i)} \text{ ou}$$

$$Y_i = \overline{G_1\ (\overline{G_{2A}} + \overline{G_{2B}})\ (C\ B\ A = i)}$$

Le décodeur 74 LS 138 ayant trois entrées d'adresse et huit sorties est appelé décodeur 3 vers 8. Il existe bien d'autres configurations de décodeur. Quelques-unes sont données en annexe de ce chapitre.

4.3. Décodage binaire – Sept segments

Pour visualiser un nombre exprimé en décimal codé binaire, on utilise couramment des dispositifs électroluminescents à sept segments contenus dans de petits boîtiers, (fig. 7.12), chacun visualisant un chiffre. Un segment est commandé par une entrée qui lui est propre (*a*, *b*, *c*, *d*, *e*, *f*, *g*). Le boîtier dispose en outre d'une entrée d'effacement de l'ensemble des segments.

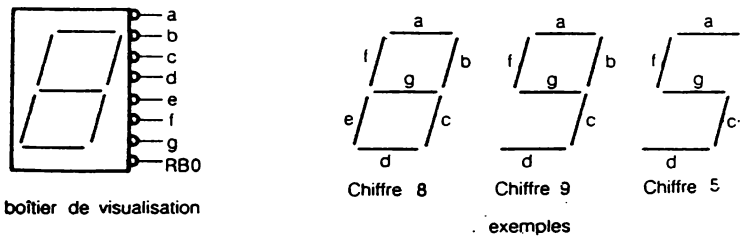


Fig. 7.12. — *Visualisation 7 segments.*

Pour activer les différents segments en fonction du code *D C B* de commande correspondant à un chiffre donné, les constructeurs proposent des décodeurs *D C B* sept segments tels que le décodeur 74 LS 247 dont le schéma symbolique est présenté à la figure 7.13.

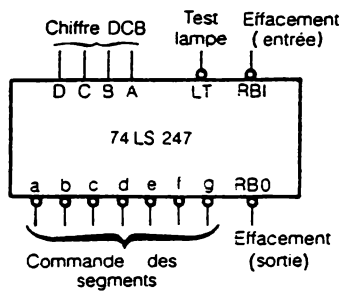


Fig. 7.13. — *Décodeur D C B sept segments.*

La table de vérité de la figure 7.14 donne le principe de fonctionnement du décodeur.

Entrées			Sorties	
Chiffre D C B	Test lampe	Effacement Entrées	Commande segments	Effacement Sorties
D C B A	LT	RBI	g f e d c b a	RBO
X X X X	0	X	1 1 1 1 1 1 1	1
X X X X	1	0	0 0 0 0 0 0 0	0
0 0 0 0	1	1	0 1 1 1 1 1 1	1
0 0 0 1	1	1	0 0 0 0 1 1 0	1
0 0 1 0	1	1	1 0 1 1 0 1 1	1
– – – –	–	–	– – – – – – –	–

1 : Segment allumé
0 : Segment éteint

Fig. 7.14. — *Table de vérité du décodeur 74 LS 247 (principe).*

4.4. Fonction d'encodage binaire prioritaire

4.4.1. Principe de l'encodage binaire

La figure 7.15 donne la table de vérité pour un encodeur à huit entrées.

Entrées									Sorties		
Validation	Code 1 parmi N								Codage binaire		
G	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	C	B	A
0	X	X	X	X	X	X	X	X	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0	1	1
1	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	1	0	1
1	0	0	0	0	0	0	1	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1	1

Fig. 7.15. — *Encodeur binaire. Table de vérité théorique.*

Les sorties A B C donnent le rang de l'entrée en 1.

L'examen de cette table amène les remarques suivantes :

1. Au niveau des entrées, elle suppose la présence d'une seule entrée active parmi N. En fait, ce cas d'exploitation se rencontre peu souvent, aussi les constructeurs réalisent-ils des encodeurs binaires prioritaires, donnant en sortie le rang de l'entrée la plus prioritaire en 1, quelque soit l'état des autres entrées ; l'ordre des priorités est fixé dans l'ordre décroissant I_7, I_6, ...
2. La combinaison de sortie $C\ B\ A = 0$ est ambigüe car elle peut signifier : information de sortie non valide, pas d'information présentée sur les entrées, entrée I_0 active. Pour lever les doutes, les constructeurs ont ajouté des sorties complémentaires.

4.4.2. Exemple : encodeur prioritaire 74 LS 348

Les figures 7.16 et 7.17 donnent respectivement le schéma symbolique de l'encodeur 74 LS 348 et sa table de vérité.

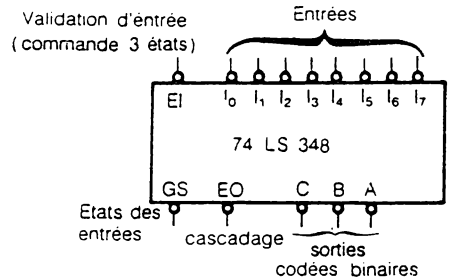


Fig. 7.16. — *Schéma symbolique de l'encodeur binaire prioritaire à sorties 3 états 74 LS 348.*

Quand l'encodeur n'est pas validé, les sorties d'adresse sont dans l'état haute impédance. Quand *EI* valide l'encodeur et s'il n'y a aucune entrée active, les sorties d'adresse restent en haute impédance. Enfin, quand l'encodeur est validé par *EI*, et qu'il y a une ou plusieurs entrées actives (au niveau 0), l'encodeur fournit en inverse sur A B C, l'adresse de l'entrée la plus prioritaire.

Exemple : I_4 active donne $C\ B\ A = 011$ ($011 = \overline{100} = \overline{4}$). La sortie *GS* indique qu'il y a une entrée active quand l'encodeur est validé et *EO* sert au

Entrées									Sorties				
EI	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A	B	C	GS	EO
1	X	X	X	X	X	X	X	X	Z	Z	Z	1	1
0	1	1	1	1	1	1	1	1	Z	Z	Z	1	0
0	X	X	X	X	X	X	X	0	0	0	0	0	1
0	X	X	X	X	X	X	0	1	0	0	1	0	1
0	X	X	X	X	X	0	1	1	0	1	0	0	1
0	X	X	X	X	0	1	1	1	0	1	1	0	1
0	X	X	X	0	1	1	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	1	0	1	0	1
0	X	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1

Fig. 7.17. — *Tableau de vérité de l'encodeur binaire prioritaire à sortie 3 états 74 LS 348.*

cascadage de boîtiers. Dans ce dernier cas, la sortie *EO* est reliée à l'entrée *EI* du boîtier suivant.

4.5. Transcodage binaire décimal

Pour effectuer un transcodage binaire décimal ou inverse, on applique un nombre en *D C B* aux entrées du circuit et il en sort l'équivalent binaire ou vice versa. Ces circuits, d'usage peu courant, sont réalisés avec des mémoires mortes (voir chapitre 15).

5. L'AIGUILLAGE D'INFORMATION

5.1. Généralités

Il s'agit d'une fonction d'usage très fréquent. La figure 7.18 montre les schémas de base d'aiguillage d'information.

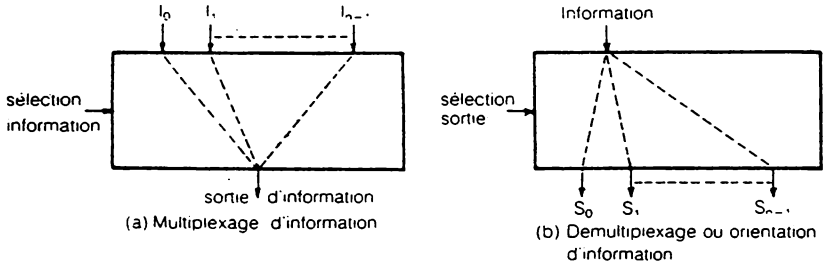


Fig. 7.18. — *Schémas d'aiguillage d'information.*

Dans le cas de la figure 7.18a, plusieurs informations arrivent à l'entrée. Par l'entrée de sélection, le circuit choisit quelle information aiguiller vers la sortie. On appelle sélecteur de données ou multiplexeur ce type de circuit.

Dans le cas de la figure 7.18b, une information est aiguillée vers une sortie choisie grâce aux entrées de sélection. Ce circuit est appelé démultiplexeur.

5.2. Circuits multiplexeur

5.2.1. Principe

La table de vérité de la figure 7.19 donne le principe d'un tel circuit et la figure 7.20 le schéma symbolique d'un multiplexeur à huit entrées.

Entrées		Sortie
Adresse	Validation	
C B A	S	Y
X X X	0	0
0 0 0	1	E_0
0 0 1	1	E_1
0 1 0	1	E_2
0 1 1	1	E_3
1 0 0	1	E_4
1 0 1	1	E_5
1 1 0	1	E_6
1 1 1	1	E_7

Fig. 7.19. — *Table de vérité de principe d'un multiplexeur 8 entrées.*

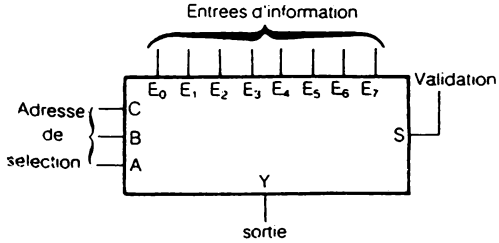


Fig. 7.20. — *Schéma symbolique d'un multiplexeur 8 entrées.*

Lorsque le multiplexeur est validé par son entrée S, la sortie Y présente l'état de l'entrée E_i sélectionnée par l'adresse $C\ B\ A = i$.

Soit :

$$Y = S.(\bar{C}\bar{B}\bar{A}E_0 + \bar{C}\bar{B}AE_1 + \bar{C}B\bar{A}E_2 + \bar{C}BAE_3 + C\bar{B}\bar{A}E_4 + C\bar{B}AE_5 + CA\bar{B}E_6 + CBAE_7)$$

ou

$$Y = S.\sum_0^7 (CBA = i).E_i$$

5.2.2. Exemple : Multiplexeur 8 entrées 74 LS 151

Les figures 7.21, 7.22, 7.23 montrent le schéma symbolique, le tableau de vérité et le schéma interne d'un multiplexeur du type 74 LS 151. Ce multiplexeur dispose de 8 entrées d'information, d'une entrée de validation S et de deux sorties complémentaires W et Y. On voit sur le tableau de vérité que lorsque le circuit est sélectionné ($S = 0$), la sortie Y reproduit l'état de l'entrée sélectionnée et W donne son inverse. Le lecteur établira facilement que :

$$Y = \bar{S}.\sum_0^7 (CBA = i).E_i \text{ et } W = \bar{Y}$$

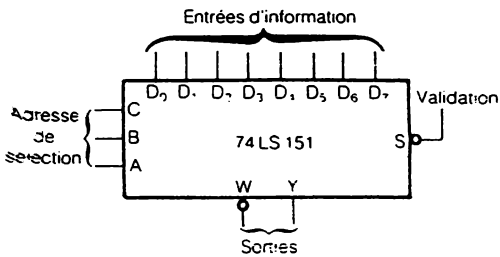


Fig. 7.21. — *Schéma symbolique du multiplexeur 74 LS 151.*

Entrées				Sorties	
Adresse			Validation		
C	B	A	S	Y	W
X	X	X	1	0	1
0	0	0	0	D_0	$\overline{D_0}$
0	0	1	0	D_1	$\overline{D_1}$
0	1	0	0	D_2	$\overline{D_2}$
0	1	1	0	D_3	$\overline{D_3}$
1	0	0	0	D_4	$\overline{D_4}$
1	0	1	0	D_5	$\overline{D_5}$
1	1	0	0	D_6	$\overline{D_6}$
1	1	1	0	D_7	$\overline{D_7}$

Fig. 7.22. — *Multiplexeur 74 LS 151 : tableau de vérité.*

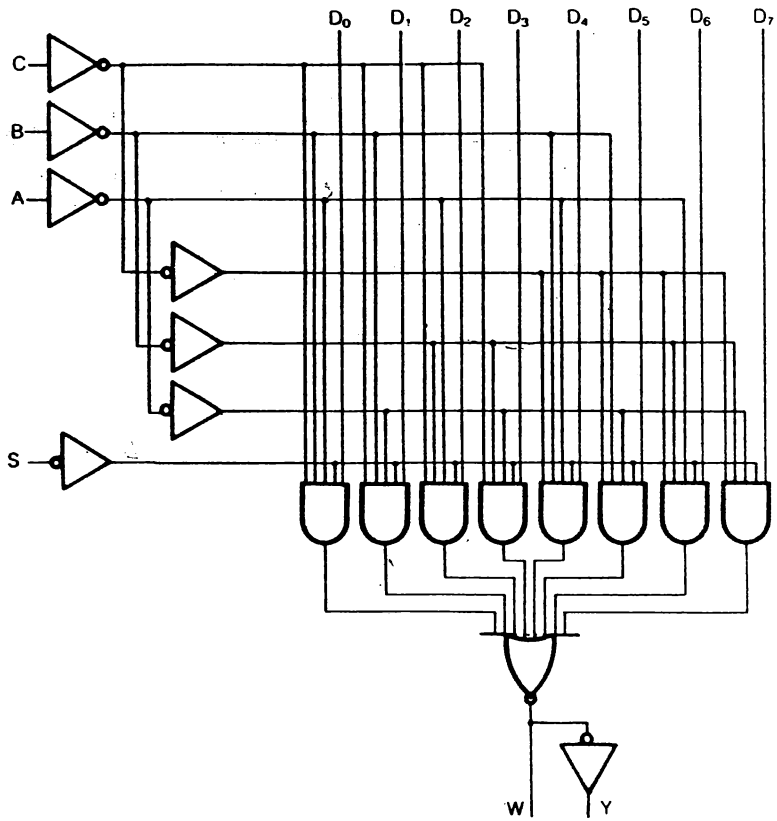


Fig. 7.23. — *Multiplexeur 74 LS 151 : schéma interne.*

Ce multiplexeur réalise l'aiguillage de 8 entrées vers une sortie. On l'appelle multiplexeur 8 vers 1. Il existe de nombreuses configurations de multiplexeurs dont quelques-unes sont données en annexe de ce chapitre.

5.3. Circuit démultiplexeur

Traduisons le fait qu'on aiguille une information I sur une sortie Y_i choisie parmi un groupe de sorties, grâce à une adresse $C\,B\,A$. Il faut que Y_i soit égal à I quand $C\,B\,A = i$.

$$Y_i = I.(C\,B\,A = i)$$

Cette équation correspond à celle de la sortie d'un décodeur $Y_i = G\,(... CBA = i)$ (voir § 4.2 de ce chapitre). C'est pourquoi les décodeurs portent aussi le nom de démultiplexeurs. Pour aiguiller une information, on la présente sur l'entrée validation du décodeur, l'adresse de sortie étant donnée par ailleurs. La figure 7.24 propose un schéma démultiplexeur 1 vers 8 utilisant un décodeur 74 LS 138.

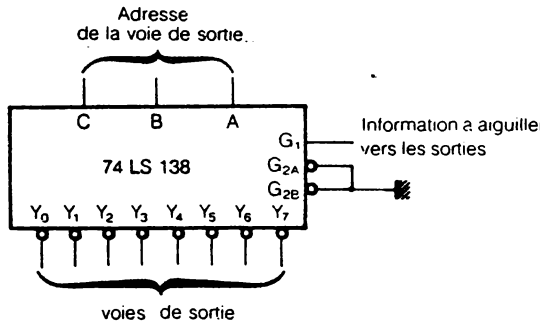


Fig. 7.24. — *Utilisation du décodeur 74 LS 138 en démultiplexeur.*

Si l'information i varie dans le temps, et que l'on souhaite aiguiller ses différents états successifs vers les sorties Y_0, Y_1, Y_2... la commande de changement d'adresse de voie devra être synchronisée dans le temps par rapport à l'évolution de l'information.

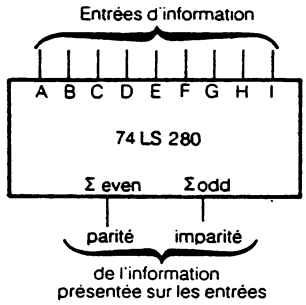
6. CALCUL ET CONTROLE DE PARITÉ : GÉNÉRATEUR DE PARITÉ

6.1. Principe

Les principes des codes détecteurs d'erreurs utilisant un eb de parité ou d'imparité ont été dégagés au chapitre 2. Nous avons vu que calcul et contrôle de parité s'effectuaient simplement avec des sommes modulo deux. Sur ces principes, les constructeurs ont intégré dans des boîtiers des sous-ensembles calculant parité et imparité sur plusieurs eb.

6.2. Le générateur de parité 74 LS 280

Le circuit 74 LS 280 calcule la parité (Σeven) ou l'imparité (Σodd) sur neuf éléments binaires (fig. 7.25).



$$\Sigma\text{even} = A \oplus B \oplus C \oplus \ldots \oplus I \text{ (Parité)}$$

$$\Sigma\text{odd} = \overline{A \oplus B \oplus C \oplus \ldots \oplus I} \text{ (Imparité)}$$

Fig. 7.25. — *Générateur de parité 74 LS 280.*

Si l'on doit calculer la parité de plus de neuf variables, on connecte la sortie Σeven à l'une des entrées d'un second boîtier, et ainsi de suite. On exploite les sorties Σeven et Σodd du dernier boîtier.

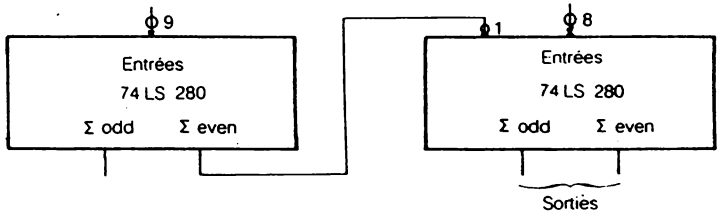


Fig. 7.26. — *Mise en cascade de générateur de parité.*

7. ÉCHANGE D'INFORMATION PAR DES BUS

Dans un système, des sous-ensembles échangent des informations comprenant plusieurs eb (8, 16 ou plus), ce qui nécessite autant de fils dans chaque sens (fig. 7.27).

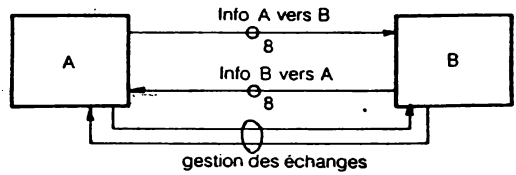


Fig. 7.27. — *Échange entre deux sous-ensembles.*

Les groupes de fils convoyant une information de même nature s'appellent des *bus*. Nous avons déjà abordé la notion de bus au chapitre 5, paragraphe 13.3.2, liés à l'utilisation de circuits trois états. Dans la figure 7.27, les fils sont *unidirectionnels* car ils ne transmettent l'information que dans un seul sens : A vers B, ou B vers A. Pour en réduire le nombre, on rend le bus bidirectionnel ; l'information allant de A vers B et B vers A emprunte les mêmes fils. Bien entendu, à un instant donné, un seul des deux échanges est possible. Examinons les organes à placer aux extrémités d'un fil, (fig. 7.28).

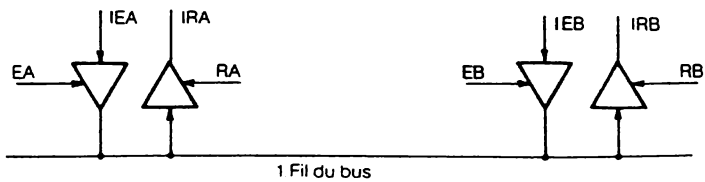
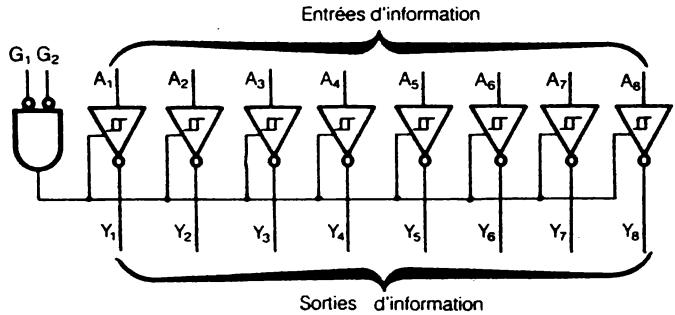


Fig. 7.28. — *Détail d'une liaison bus bidirectionnel.*

Dans le sens *A* vers *B*, l'information à émettre *IEA* est placée à l'entrée d'un "tri-state" et émise sur le bus par un signal *EA*. De l'autre côté, le récepteur est validé par *RB* et l'on obtient l'information reçue *IRB*. Le schéma est symétrique dans l'autre sens. Les constructeurs ont intégré ces différentes portes "tri-state" pour gérer plusieurs fils à la fois. Les boîtiers à 20 broches permettent de réaliser des circuits traitant huit éléments binaires.

Le circuit le plus simple est l'amplificateur de bus ou "bus buffer" ou "line driver" unidirectionnel. La figure 7.29 montre le schéma interne du 74 LS 540, constitué de 8 portes trois états, validées par deux commandes G_1 et G_2.



$G_1 + G_2$	Y_i
1	haute impédance
0	A_i

Fonction

Fig. 7.29. — *Schéma du 74 LS 540 (bus buffer).*

Pour réaliser une liaison bidirectionnelle avec ce type de circuit, il suffit d'en placer deux tête-bêche et de commander de manière exclusive les ouvertures et fermetures (fig. 7.30). Les constructeurs fournissent maintenant, tout intégré, des circuits de ce type. La figure 7.31 montre le schéma du récepteur émetteur de bus dit "bus transceiver" 74 LS 640. L'entrée *DIR* détermine le sens de transfert de l'information *A* vers *B* ou *B* vers *A*, tandis que l'entrée *G* valide en mode basse impédance le sens choisi ou bloque les deux sorties en mode haute impédance, isolant ainsi les bus *A* et *B*.

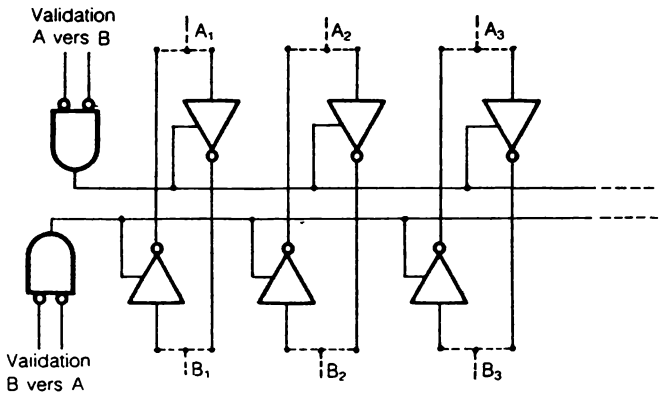


Fig. 7.30. — *Réalisation d'un coupleur de bus bidirectionnel avec deux 74 LS 540.*

Les circuits de gestion de bus sont prévus pour alimenter un grand nombre de charges. En conséquence, ils délivrent des courants I_{OH} et I_{OL} élevés. Par exemple pour le 74 LS 640.

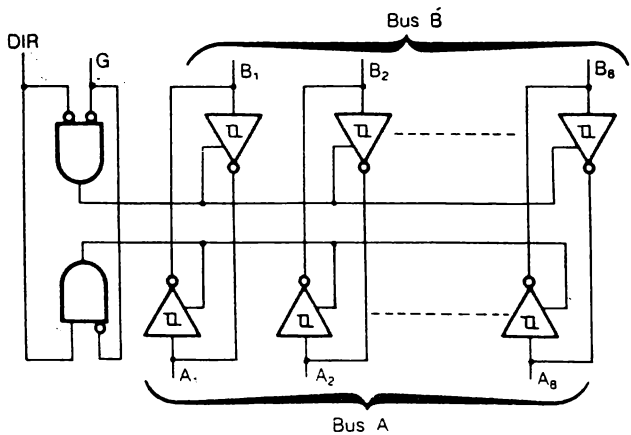
$I_{OH} = -15\ mA$ ($-400\ \mu A$ pour une porte *LS* standard)

$I_{OL} = +24\ mA$ ($+8mA$ pour une porte *LS* standard).

A l'état haute impédance, les sorties se comportent comme des entrées de portes.

$I_{OZH} = +\ 20\ \mu A$ ($+20\ \mu A$ pour la porte 3 états 74 LS 125)

$I_{OZL} = -400\ \mu A$ ($-20\ \mu A$ pour la porte 3 états 74 LS 125).



G	DIR	Bus A	Bus B
1	X	Sorties haute impédance	
0	0	$\overline{A}$	
0	1		$\overline{B}$

Fonction

Fig. 7.31. — *Schéma du coupleur de bus bidirectionnel 74 LS 640.*

Les portes trois états possèdent en outre des entrées de type trigger de Schmitt pour améliorer l'immunité au bruit. Ces circuits de gestion de bus s'utilisent principalement dans les systèmes à microprocesseurs (circuits VLSI de traitement d'information) pour permettre les échanges entre sous-ensembles fonctionnels.

8. CIRCUITS ARITHMÉTIQUES

8.1. Équations logiques d'un additionneur

Étant donné deux nombres A et B à n éléments binaires, la somme Σ de ces deux nombres s'obtient suivant le principe illustré à la figure 7.32.

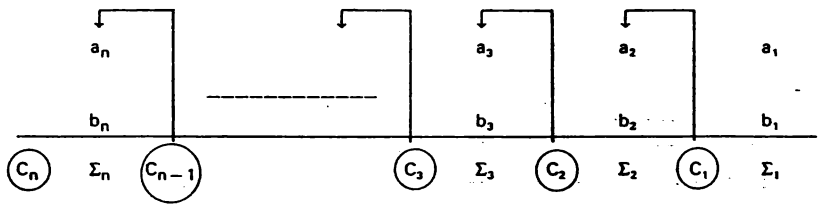


Fig. 7.32. — *Addition de deux nombres binaires.*

Soit les termes de rang i a_i et b_i, la somme Σ_i s'obtient à partir des termes a_i et b_i et de la retenue du rang précédent c_{i-1}. Ces mêmes termes permettent de calculer la retenue C_i appliquée au rang $i+1$.

Un additionneur peut donc être réalisé par la juxtaposition de sous-ensembles combinatoires identiques calculant la somme Σ_i et la retenue C_i (fig. 7.33) à partir des variables a_i, b_i et C_{i-1}.

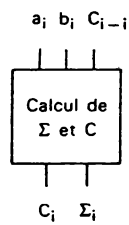


Fig. 7.33. — *Étage d'un additionneur.*

La table de vérité permettant de déterminer les fonctions Σ_i et C_i en fonction des variables a_i, b_i, C_{i-1} est donnée à la figure 7.34. Elle s'établit sans difficulté. On tire de la table de vérité les équations logiques des fonctions Σ_i et C_i en s'aidant des tableaux de Karnaugh de la figure 7.35.

a_i	b_i	C_{i-1}	Σ_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig. 7.34. — *Table de vérité d'un additionneur.*

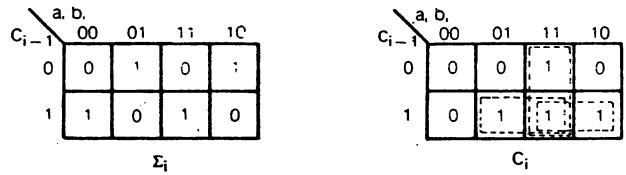


Fig. 7.35. — *Simplification des fonctions Σ_i et C_i.*

① $\Sigma_i = \bar{a}_i \bar{b}_i C_{i-1} + \bar{a}_i b_i \overline{C_{i-1}} + a_i b_i C_{i-1} + a_i \bar{b}_i \overline{C_{i-1}} = a_i \oplus b_i \oplus C_{i-1}$

② $C_i = a_i b_i + a_i C_{i-1} + b_i C_{i-1} = a_i b_i + C_{i-1}.(a_i + bi)$

Ou encore sans effectuer la simplification maximum de la fonction C_i:

③ $C_i = a_i b_i + \bar{a}_i b_i C_{i-1} + a_i \bar{b}_i C_{i-1} = a_i b_i + C_{i-1}.(a_i \oplus b_i)$.

8.2. Réalisation d'un additionneur à retenue série

La figure 7.36 donne le schéma d'un additionneur réalisé en appliquant les formules ① et ③. La retenue est dite à propagation série car une retenue entrant en C_0, en provenance de l'étage précédent, ne parvient en C_4 pour attaquer l'étage suivant qu'après avoir traversé 4 portes ET et 4 portes OU. Une réalisation avec des portes individuelles de la série 74 LS conduirait à un retard de 170 *ns* environ.

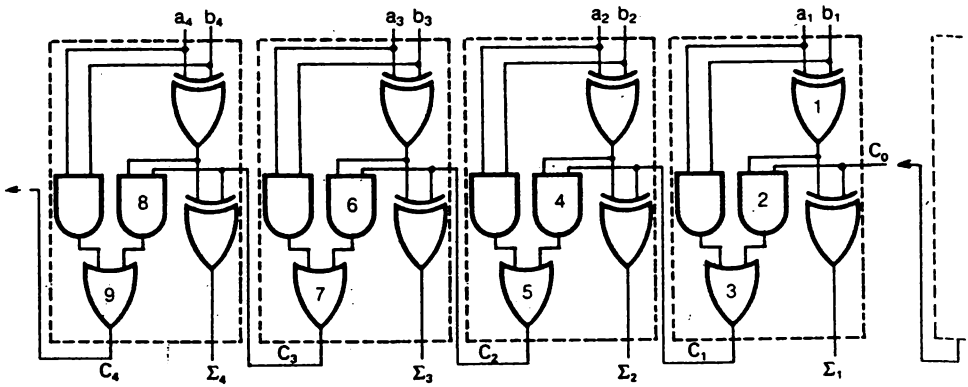


Fig. 7.36. — *Schéma de principe d'un additionneur à 4 eb.*

La figure 7.37 montre un des premiers additionneurs à retenue série, le 7482, qui n'est plus d'un usage courant maintenant, car dépassé par d'autres boîtiers plus performants.

Ce circuit calcule de façon originale la somme et la retenue. Il ne correspond pas du tout aux formules du paragraphe 8.1. La somme Σ_i dépend de C_0, A_1, B_1 et de $\overline{C}_1$ (la théorie implique seulement C_0, A_1, B_1) et Σ_2 dépend de $\overline{C}_1$, $\overline{A}_2$, $\overline{B}_2$, C_2. On peut vérifier cependant que le circuit réalise bien les fonctions attendues. En technologie TTL standard, ce circuit donnait un retard de 42 *ns* sur la retenue pour deux étages (84 *ns* pour 4).

8.3. Additionneur à retenue anticipée: 74 LS 83 A

Pour augmenter les performances de l'additionneur, on calcule à chaque étage la retenue C_i en fonction de $a_i\, b_i$ et de tous les étages précédents a_{i-1}, b_{i-1}, a_{i-2}, b_{i-2}, ... a_1, b_1. Ceci évite d'attendre que C_{i-1} soit élaborée, retenue dépendant de C_{i-2}, etc... On dit que la retenue est anticipée (look-ahead carry). La figure 7.38 montre le schéma symbolique correspondant. Reprenons la formule de la retenue C_i et montront qu'elle peut s'exprimer en fonction des bits à additionner et de la retenue entrante.

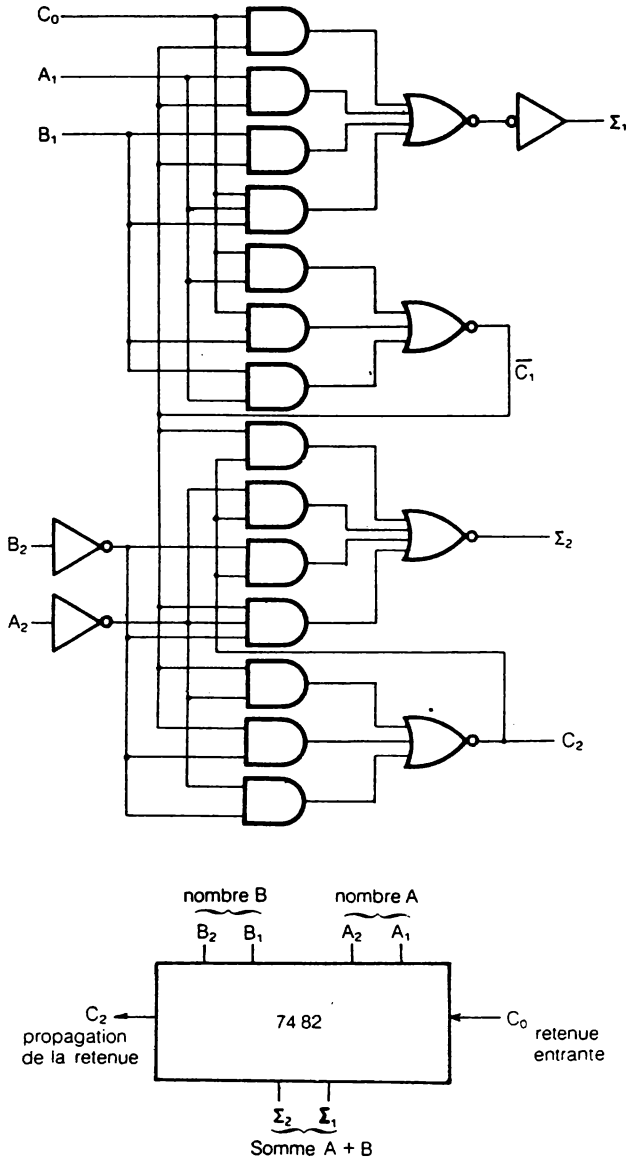


Fig. 7.37. — *Additionneur sur 2 eb 7482.*

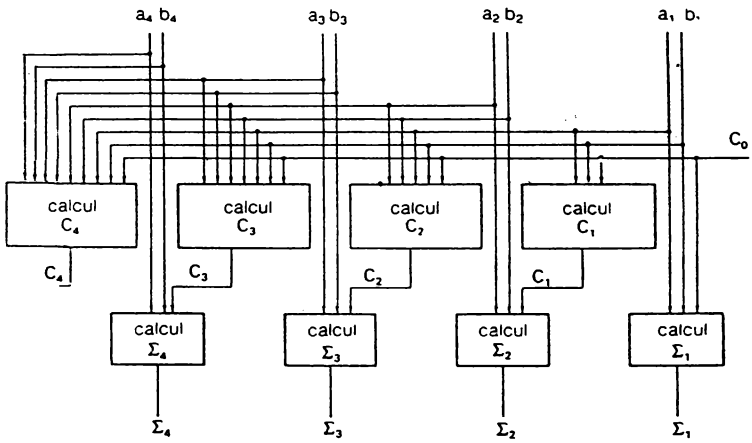


Fig. 7.38. — *Principe d'un additionneur à retenue anticipée.*

$$C_i = a_i\, b_i + a_i\, C_{i-1} + b_i\, C_{i-1} = a_i\, b_i + (a_i + b_i).C_{i-1}$$

$C_i = G_i + P_i\, C_{i-1}$ en posant $G_i = a_i\, b_i$ et $P_i = a_i + b_i$.

Le terme G_i est appelé terme de *génération* de retenue car s'il vaut 1, c'est-à-dire si a_i et b_i valent 1 simultanément, une retenue est générée, indépendamment de la retenue des étages précédents.

Le terme P_i est appelé terme de *propagation* de retenue car s'il vaut 1, c'est-à-dire si a_i ou b_i valent 1, la retenue des étages précédents est propagée.

L'équation $C_i = G_i + P_i\, C_{i-1}$ exprime que l'étage i fournit une retenue s'il en génère une ou qu'il propage celle de l'étage précédent. On en déduit aisément l'expression de la retenue pour chacun des étages à partir de la retenue initiale C_0.

$$C_1 = G_1 + P_1\, C_0$$

$$C_2 = G_2 + P_2\, C_1 = G_2 + P_2\, G_1 + P_2\, P_1\, C_0$$

$$C_3 = G_3 + P_3\, C_2 = G_3 + P_3\, G_2 + P_3\, P_2\, G_1 + P_3\, P_2\, P_1\, C_0$$

$$C_4 = G_4 + P_4\, C_3 = G_4 + P_4\, G_3 + P_4\, P_3\, G_2 + P_4\, P_3\, P_2\, G_1 + P_4\, P_3\, P_2\, P_1\, C_0.$$

On voit ainsi que chacune des retenues ne dépend que des nombres à additionner par l'intermédiaire des fonctions P et G et de la retenue entrante

C_0. On remplace une propagation série par une propagation parallèle: toutes les retenues sont calculées en même temps.

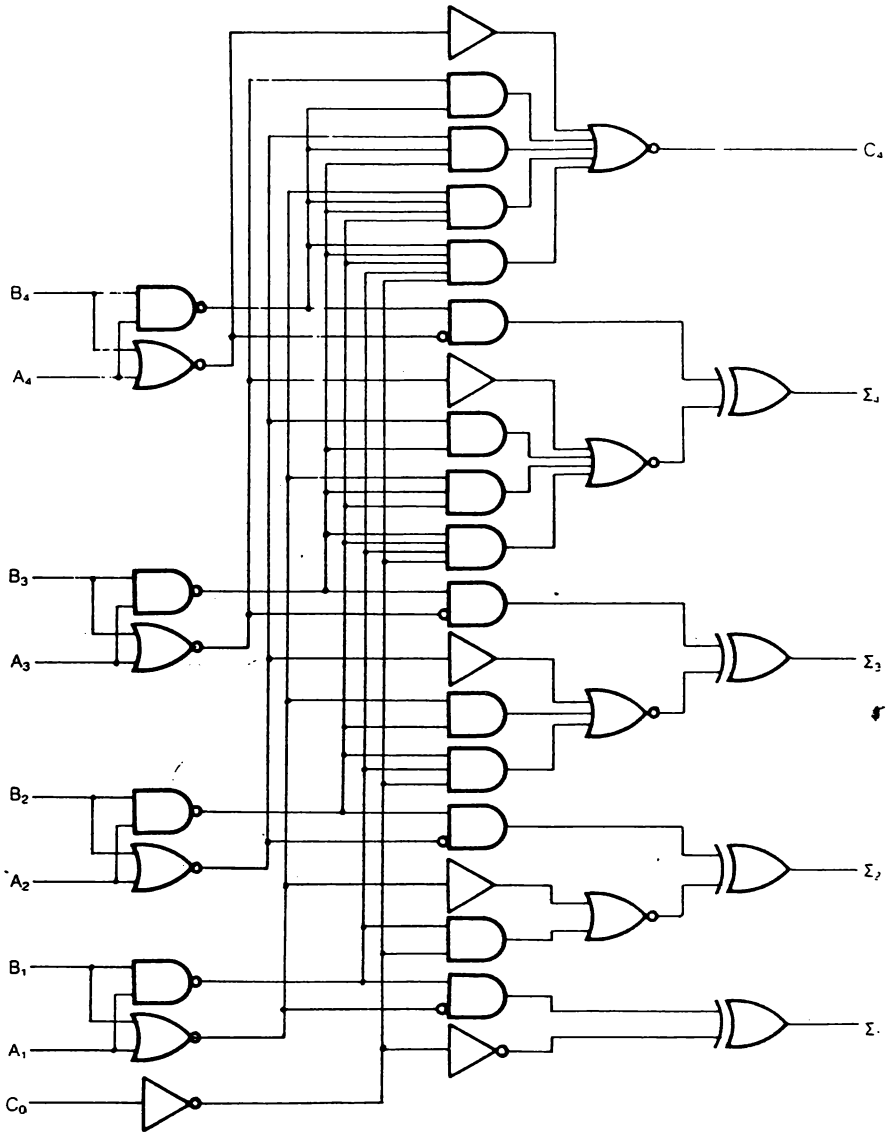


Fig. 7.39. — *Additionneur à retenue anticipée 74 LS 83 A.*

Dans le pire des cas, le temps de propagation de la retenue C_4 du 74 LS 83 A est de 24 *ns*, à rapprocher des valeurs précédemment trouvées, 170 *ns* et 84 *ns*. Cela montre tout l'intérêt des additionneurs à calcul de retenue anticipée. Cette augmentation de la rapidité s'accompagne cependant d'une augmentation sensible de la complexité interne.

8.4. Bloc de calcul arithmétique et logique

La figure 7.40 donne la table de vérité des blocs de calcul 74 LS 381 et 74 LS 382. Ces deux circuits reçoivent deux opérandes A et B de 4 eb chacun, et en fonction de 3 eb de commande S_0, S_1, S_2, exécutent l'une des 8 fonctions prévues dans le tableau. Par exemple, pour $S_2\,S_1\,S_0 = 101$, la sortie F prend la valeur $A \oplus B$.

S_2	S_1	S_0	Opération réalisée	
0	0	0	F = 0000	
0	0	1	F = B moins A	Opérateurs arithmétiques
0	1	0	F = A moins B	
0	1	1	F = A plus B	
1	0	0	F = A · B	Opérateurs logiques
1	0	1	F = A ⊙ B	
1	1	0	F = A B	
1	1	1	F = 1111	

Fig. 7.40. — *Fonctions réalisées par les circuits de calcul 74 LS 381 et 74 LS 382.*

Ces deux blocs diffèrent l'un de l'autre par le mode de calcul de la retenue. Le 74 LS 382 (fig. 7.41) calcule les retenues en parallèle à l'intérieur du boîtier et propage la retenue du dernier étage interne vers le boîtier suivant. Pour un additionneur à 16 étages, le temps de calcul global de la retenue sera donc égal à quatre fois celui d'un boîtier. Ce même boîtier calcule une fonction de débordement de capacité, telle que nous l'avons définie au paragraphe 4.2 du chapitre 4.

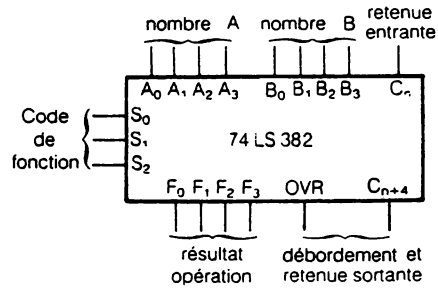


Fig. 7.41. — *Bloc de calcul 74 LS 382.*

Le circuit 74 LS 381 (fig. 7.42) fournit deux sorties de propagation et de génération de retenue P et G qui, reliées aux entrées correspondantes d'un circuit 74 S 182 calcule les retenues anticipées entièrement en parallèle.

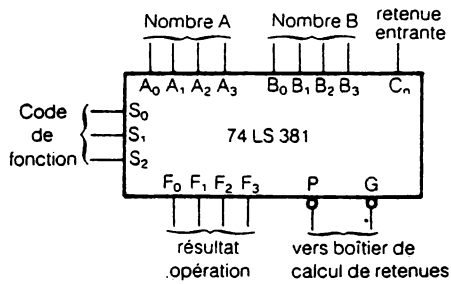


Fig. 7.42. — *Bloc de calcul 74 LS 381 à calcul de retenues anticipées.*

8.5. Multiplieur

Au chapitre 3, nous avons montré le principe de la multiplication des nombres binaires non signés ou signés en complément à 2. La figure 7.43 montre un multiplieur parallèle réalisant la multiplication des deux nombres signés ou non signés X et Y de 8 eb chacun pour donner un résultat signé ou non signé de 16 eb (type 67558 de Monolithic Memories Inc).

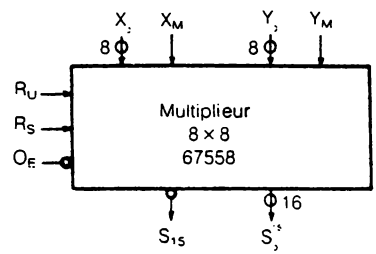


Fig. 7.43. — *Multiplieur parallèle 67558 de MMI.*

Les entrées X_M et Y_M précisent si le nombre correspondant X et Y est signé ou non, tandis que les entrées R_u et R_s permettent d'obtenir une valeur arrondie du produit pour des nombres non signés et signés. La faculté d'opérer sur des nombres signés et non signés sert quand on réalise des multiplications sur plus de 8 eb. Seule la partie de poids la plus élevée doit être considérée comme signée, les parties de poids plus faible sont considérées comme non signées.

Ce multiplieur, très rapide, fournit le produit 150 *ns* après application des nombres X, Y sur les entrées.

9. PARAMÈTRES DES CIRCUITS COMBINATOIRES COMPLEXES

On retrouve pour les circuits combinatoires complexes les mêmes paramètres électriques et dynamiques que pour les portes simples. Mais la variété des entrées agissant sur une même sortie fait que les paramètres dynamiques sont plus nombreux. L'utilisateur fera attention aux entrances particulières de certaines entrées qui consomment parfois plus de courant qu'une entrée de porte standard, ou parfois moins. A titre d'exemple, examinons les paramètres du circuit comparateur 74 LS 85 (fig. 7.44).

La figure 7.45 donne les paramètres électriques de ce circuit. En comparant à ceux du boîtier standard de la famille, le 74 LS 00 à 4 portes NAND, on note que les entrées $A < B$ et $A > B$ se comportent normalement, tandis que toutes les autres absorbent ou fournissent trois fois la valeur normale du courant. Dans les schémas utilisant ce circuit comparateur, il faudra donc compter comme trois charges toutes les entrées sauf

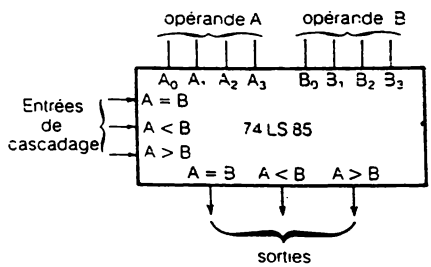


Fig. 7.44. — *Comparateur 74 LS 85.*

Paramètres		TYP MIN	NOM	MAX	Unité	Comparaison avec porte standard 74 LS 00
Tension d'alimentation V_{CC}		4,75	5	5,25	V	Idem
Courant de sortie I_{OH}				−400	µA	Idem
Température de fonctionnement T_A		0		70	°C	Idem
Tension d'entrée V_{IH}		2			V	Idem
Tension d'entrée V_{IL}				0,8	V	Idem
Tension d'écrêtage V_{IK}				−1,5	V	Idem
Tension de sortie V_{OH}		2,7	3,4		V	Idem
Tension de sortie V_{OL} (I_{OL}=8 mA)			0,35	0,5	V	0,25 en typique
Courant maximum d'entrée I_I	Entrées A < B et A > B			0,1	mA	Idem
	Autres entrées			0,3	mA	3 fois la valeur standard
Courant d'entrée I_{IH}	Entrées A < B et A > B			20	µA	Idem
	Autres entrées			60	µA	3 fois la valeur standard
Courant d'entrée I_{IL}	Entrées A < B et A > B			−0,4	mA	Idem
	Autres entrées			−1,2	mA	3 fois la valeur standard
Courant de court-circuit I_{OS}		−20		−100	mA	Idem
Courant d'alimentation I_{CC}			10,4	20	mA	4,4 mA max pour boîtier 74 LS 00

Fig. 7.45. — *Paramètres électriques du comparateur 74 LS 85.*

$A < B$ et $A > B$. Les tensions d'entrées et de sorties sont évidemment les mêmes pour assurer la compatibilité des circuits. Les courants de sortie sont identiques également.

Pour les paramètres dynamiques, la figure 7.46 montre que les retards sont spécifiés entre une entrée donnée et une sortie donnée et que le retard dépend du chemin emprunté dans les portes internes au boîtier. Certains paramètres ne figurent pas dans le pire des cas, mais seulement en typique. Si le montage exploite les sorties correspondantes, il faudra prendre le pire des cas pour les sorties spécifiées.

Paramètres	De l'entrée	A la sortie	Nombre de couches traversées	MIN	TYP	MAX	UNITE
t_{PLH}	N'importe quel A ou B	A < B, A > B	1		14		ns
			2		19		ns
			3		24	36	ns
		A = B	4		27	45	ns
t_{PHL}	N'importe quel A ou B	A < B, A > B	1		11		ns
			2		15		ns
			3		20	30	ns
		A = B	4		23	45	ns
t_{PLH}	A < B ou A = B	A > B	1		14	22	ns
t_{PHL}	A < B ou A = B	A > B	1		11	17	ns
t_{PHL}	A = B	A = B	2		13	26	ns
t_{PLH}	A > B ou A = B	A < B	1		14	22	ns
t_{PHL}	A > B ou A = B	A < B	1		11	17	ns

Fig. 7.46. — *Paramètres dynamiques du comparateur 74 LS 85.*

D'une manière générale, le concepteur n'utilise pas tous les paramètres figurant dans ce tableau mais seulement les valeurs maximum. Ainsi, sur le montage de la figure 7.47 qui montre plusieurs comparateurs montés en cascade, le temps maximum d'établissement des sorties $A = B$, $A > B$, $A < B$ est égal à 45 *ns* plus $(n - 1)$ 26 *ns*, n étant le nombre de boîtiers montés en cascade. Ce temps est compté à partir de la stabilisation des entrées d'information, les entrées $A < B$, $A > B$, $A = B$ du premier boîtier de la chaîne étant placées à un niveau fixe.

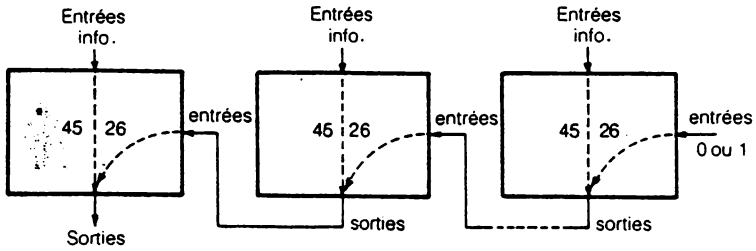


Fig. 7.47. — *Cumul des retards dans une chaîne.*

ANNEXE DU CHAPITRE 7

1. OPÉRATEURS DE BASE : PRINCIPAUX CIRCUITS

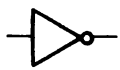
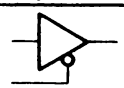
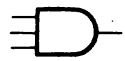
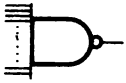
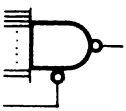
Type d'opérateur	Symbole	Nombre d'opérateurs par boîtier	Référence boîtier	Collecteur ouvert	Amplificateur		Trigger de Schmitt	observations
					Tension de sortie > 5 V	Fort courant		
Inverseur		6	74 LS 04					
			74 LS 05	X				
			74 LS 06	X	30 V	X		
			74 LS 16	X	15 V	X		
			74 LS 14				X	
Séparateur		6	74 LS 07	X	30 V	X		
			74 LS 17	X	15 V	X		
Séparateur 3 états		4	74 LS 125					
		4	74 LS 126A					
AND		4	74 LS 08					
			74 LS 09	X				
		3	74 LS 11					
			74 LS 15	X				
OR		4	74 LS 32					
NAND		4	74 LS 00					
			74 LS 01	X				brochages différents
			74 LS 03	X				
			74 LS 26	X	15 V			
			74 LS 37			X		
			74 LS 38	X		X		
			74 LS 132				X	
NAND		3	74 LS 10					
			74 LS 12	X				
		2	74 LS 20					
			74 LS 22	X				
			74 LS 40			X		
		1	74 LS 30					8 entrées
			74 LS 133					13 entrées
		1	74 LS 134					12 entrées Sortie 3 états

Fig. 7.48(a).

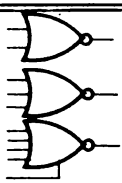
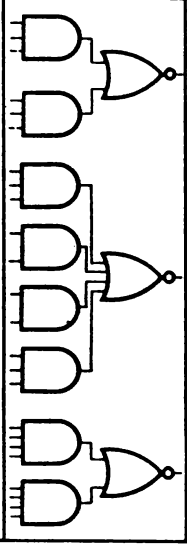
Type d'opérateur	Symbole	Nombre d'opérateurs par boîtier	Référence boîtier	Collecteur ouvert	Amplificateur : Tension de sortie > 5 V	Amplificateur : Fort courant	Trigger de Schmitt	observations
NOR		4	74 LS 02 74 LS 28			X		
		3	74 LS 33 74 LS 27	X				
		2	74 LS 25					avec échan-tillonnage
Somme modulo 2		4	74 LS 86 74 LS 386 74 LS 136 74 LS 266	 X X				brochages différents brochages différents
Aiguillage		2	74LS51					opérateurs a portes ET à 2 ou 3 entrées
		1	74LS54					
		1	74LS55					

Fig. 7.48(b).

2. FONCTIONS COMBINATOIRES USUELLES PRINCIPAUX TYPES DE CIRCUITS MSI

2.1. Comparaison entre information

2.1.1. Comparateur cascadable de deux nombres de 8 eb (AM 25 LS 2521) boîtier 24 broches, présenté au paragraphe 3.2.2.

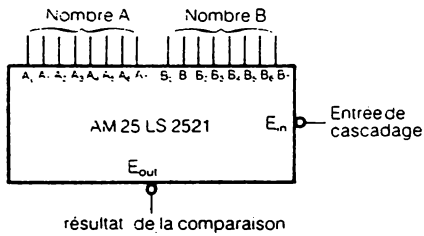


Fig. 7.49

$$E_{out} = \overline{\overline{E_{in}}(\overline{A_0 \oplus B_0})\,(\overline{A_1 \oplus B_1}) \ldots (\overline{A_7 \oplus B_7})} = \overline{\overline{E_{in}} \cdot (A = B)}$$

2.1.2. Comparateur cascadable de deux nombres de 4 eb (74 LS 85). Boîtier 16 broches présenté au paragraphe 3.3.3.

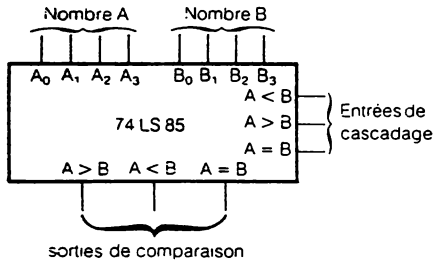


Fig. 7.50.

2.1.3. Comparateur non cascadable de deux nombres de 8 eb (74 LS 682). Boîtier 20 broches.

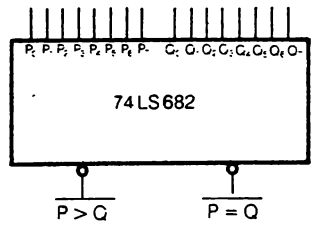


Fig. 7.51.

2.1.4. Comparateur cascadable de deux nombres de 8 eb (74 LS 686). Boîtier de 24 broches.

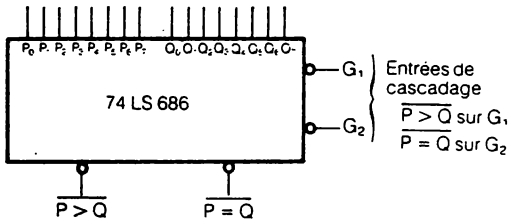
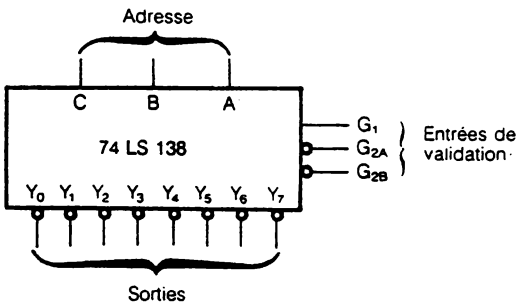


Fig. 7.52.

2.2. Décodeurs (binaire vers 1 parmi N)

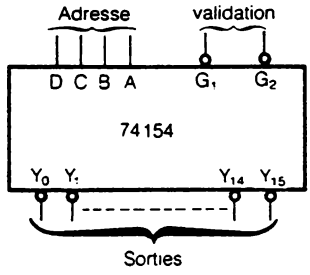
2.2.1. Décodeur 8 sorties avec entrée de validation (74 LS 138). Boîtier 24 broches présenté au paragraphe 4.2.2.



$$Y_1 = \overline{G_1(\overline{G_2 A + G_2 B})\,(CBA = i)}$$

Fig. 7.53.

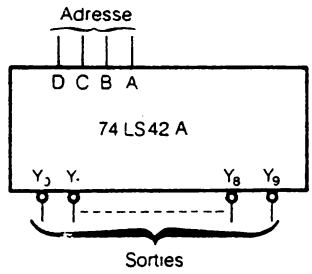
2.2.2. Décodeur 16 sorties avec entrée de validation (74 154). Boîtier 24 broches.



$$Y_i = \overline{\bar{G}_1\,\bar{G}_2\,(DCBA = i)}$$

Fig. 7.54.

2.2.3. Décodeur 10 sorties (74 LS 42 A). Boîtier 16 broches.

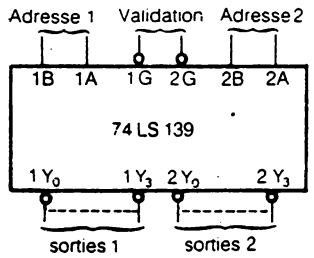


$$\overline{Y_i = (DCBA = i)} \text{ pour } 0 \leqslant i \leqslant 9$$

$$Y_i = 1 \qquad \text{pour } 10 \leqslant i \leqslant 15$$

Fig. 7.55.

2.2.4. Double décodeur à 4 sorties avec entrées de validation (74 LS 139). Boîtier 16 broches. Deux décodeurs indépendants dans le boîtier.

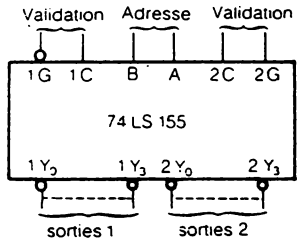


$$1Y_{i_1} = \overline{\overline{1G}\,(1B1A = i_1)}$$

$$2Y_{i_2} = \overline{\overline{2G}\,(2B2A = i_2)}$$

Fig. 7.56.

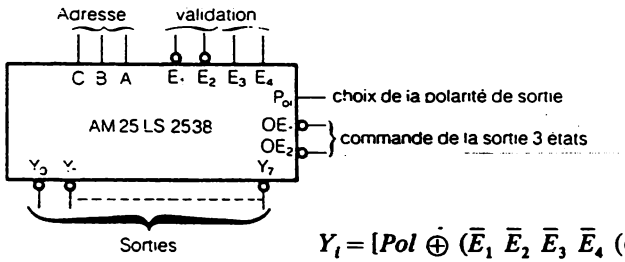
2.2.5. Double décodeur à adresse unique et double validation (74 LS 155). Boîtier 16 broches.



$$1Y_i = \overline{\overline{1G}\ 1C\,(BA = i)}$$
$$2Y_i = \overline{2G\ 2C\,(BA = i)}$$

Fig. 7.57.

2.2.6. Décodeurs à 8 sorties 3 états, avec validation des sorties et choix de la polarité de sortie des signaux (boîtier 20 broches).



$$Y_i = [Pol \oplus (\overline{E}_1\ \overline{E}_2\ \overline{E}_3\ \overline{E}_4\ (CBA = i))]\ \overline{OE}_1\ \overline{OE}_2$$

Fig. 7.58.

2.3. Décodeur DCB sept segments (74 LS 246)

Boîtier 16 broches, présenté au paragraphe 4.3.

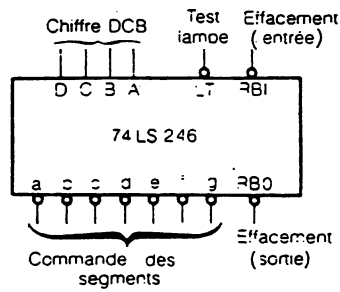


Fig. 7.59.

2.4. Encodeur binaire prioritaire (74 LS 348)

Boîtier 16 broches présenté au paragraphe 4.4.2.

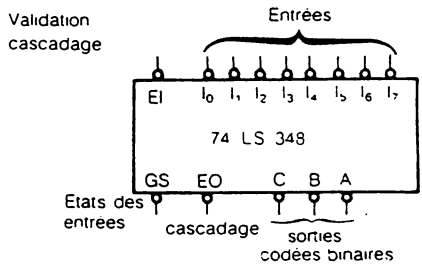
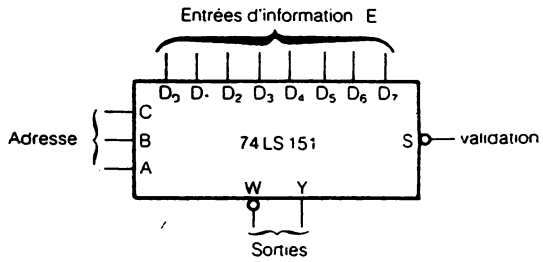


Fig. 7.60.

2.5. Multiplexeurs

2.5.1. Multiplexeur 8 vers 1 (74 LS 151). Boîtier 16 broches présenté au paragraphe 5.2.2.

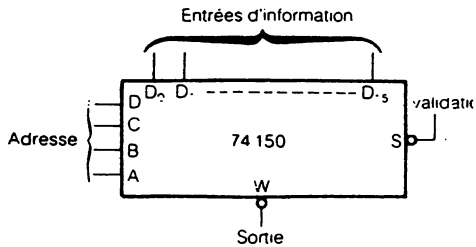


$$Y = \overline{S} \sum_{i=0}^{7} (CBA = i)\, D_i$$

$$W = \overline{Y}$$

Fig. 7.61.

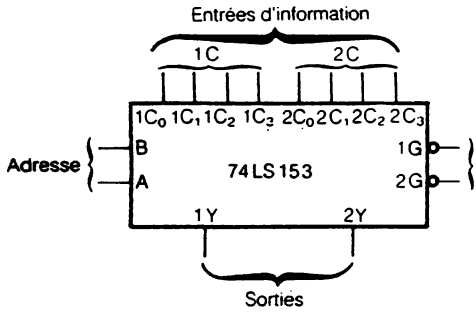
2.5.2. Multiplexeur 16 vers 1 (74 150). Boîtier 24 broches.



$$W = \overline{S} \sum_{i=0}^{15} (CBA = i)\, D_i$$

Fig. 7.62.

2.5.3. Double multiplexeur 4 vers 1 (74 LS 153). Boîtier 16 broches.

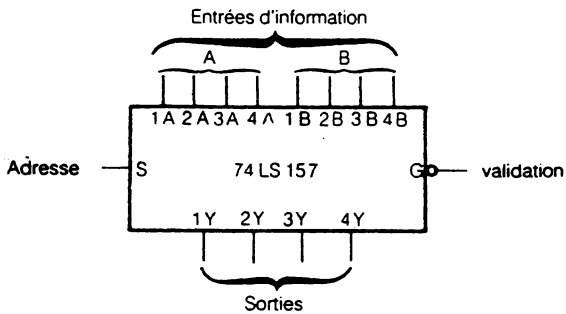


$$1Y = \overline{1G} \sum_{i=0}^{3} (BA = i)\, 1C_i$$

$$2Y = \overline{2G} \sum_{i=0}^{3} (BA = i)\, 2C_i$$

Fig. 7.63.

2.5.4. Quadruple multiplexeur 2 vers 1 (74 LS 157). Boîtier 16 broches.

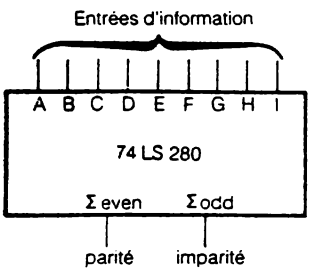


$$1Y = \bar{G}.(\bar{S}.1A + S.1B)$$
$$2Y = \bar{G}.(\bar{S}.2A + S.2B)$$
$$3Y = \bar{G}.(\bar{S}.3A + S.3B)$$
$$4Y = \bar{G}.(\bar{S}.4A + S.4B)$$

Fig. 7.64.

2.6. Générateur de parité (74 LS 280)

Boîtier 16 broches présenté au paragraphe 6.2.

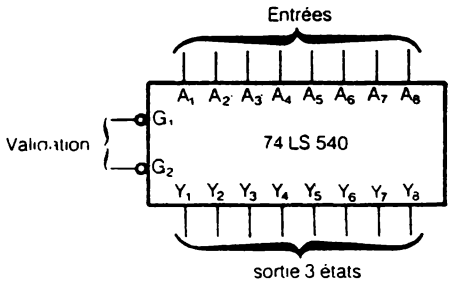


$$\Sigma\text{even} = A \oplus B \oplus C \ldots \oplus I \text{ (Parité)}$$
$$\Sigma\text{odd} = \overline{A \oplus B \oplus C \ldots \oplus I} \text{ (Imparité)}$$

Fig. 7.65.

2.7. Commandes de bus trois états

2.7.1. Commande de bus 3 états (74 LS 540). Boîtier 20 broches présenté au paragraphe 7.

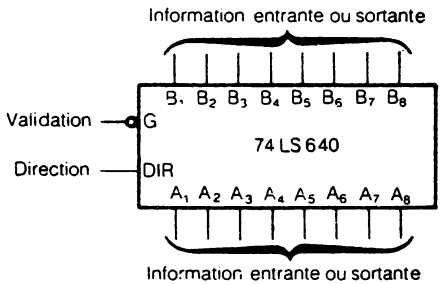


$G_1 + G_2$	Y
1	haute impédance
0	A_i

$$Y_i = (\overline{G_1 + G_2}).\bar{A}_i$$

Fig. 7.66.

2.7.2. Commande de bus 3 états bidirectionnel (74 LS 640) présenté au paragraphe 7.



$$A_i = \bar{G}.\overline{DIR}.\bar{B}_i$$

$$B_i = \bar{G}.\overline{DIR}.\bar{A}_i$$

Fig. 7.67.

2.8. Circuits arithmétiques

2.8.1. Additionneur à retenue série (7482) présenté au paragraphe 8.2.

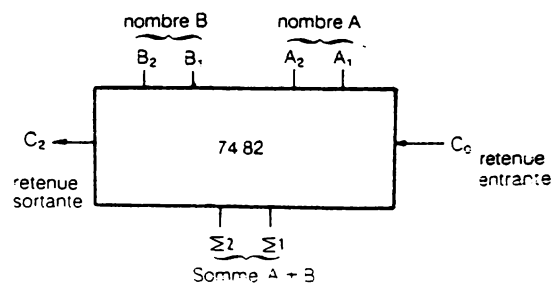


Fig. 7.68.

2.8.2. Additionneur à retenue parallèle (74 LS 83 A) présenté au paragraphe 8.3.

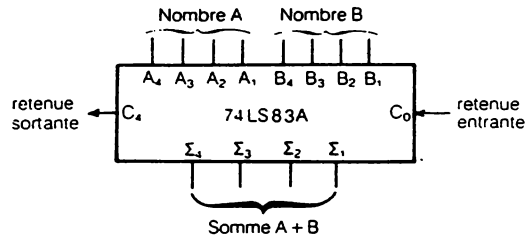


Fig. 7.69.

2.8.3. Bloc de calcul arithmétique (74 LS 381) et (74 LS 382) présenté au paragraphe 8.4.

S_2	S_1	S_0	Opération réalisée
0	0	0	F = 000
0	0	1	F = B moins A
0	1	0	F = A moins B
0	1	1	F = A plus B
1	0	0	F = A + B
1	0	1	F = A ⊕ B
1	1	0	F = A B
1	1	1	F = 111

Table de vérité commune

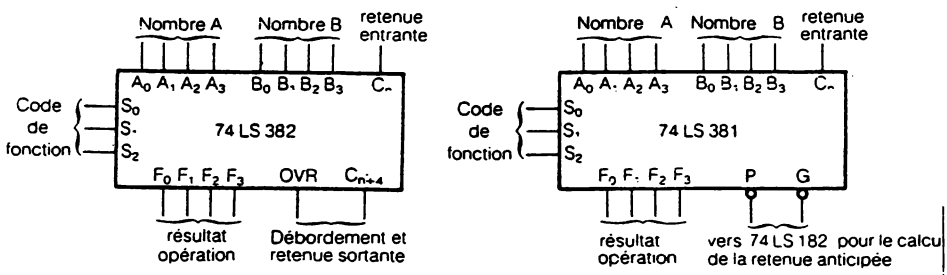


Fig. 7.70.

2.8.4. Multiplieur (AM 2505)

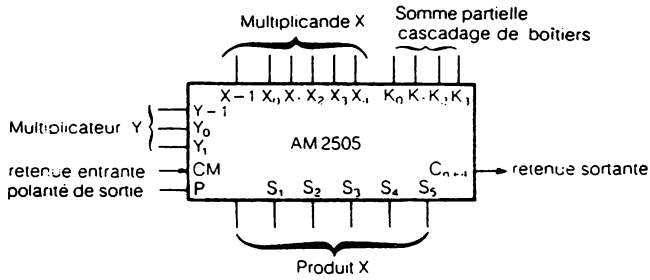


Fig. 7.71.

2.8.5. Multiplieur MMI 67558 présenté au paragraphe 8.5.

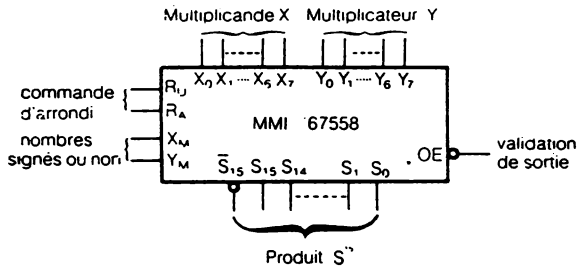


Fig. 7.72.

Chapitre 8

EXTENSION DES FONCTIONS COMBINATOIRES

1. LES EXTENSIONS DES FONCTIONS COMBINATOIRES

Le nombre limité des broches d'un boîtier (14, 16, 20, 24 pour les plus courants) fait qu'on ne peut calculer que quelques fonctions de quelques variables seulement. Si l'on a affaire à beaucoup de fonctions dépendant de nombreuses variables, il faut associer plusieurs boîtiers. Pour certains d'entre eux, l'association est prévue par principe à la construction. Le boîtier possède alors des entrées dites de "cascadage" ou de "mise en cascade".

2. LES EXTENSIONS DES FONCTIONS ET, OU

Les boîtiers NAND et NOR les plus courants ont 2, 3 ou 4 fonctions de 2, 3 ou 4 variables chacune. Si l'on veut calculer par exemple une fonction ET ou bien une fonction OU de 16 variables, il faut procéder par association. Soit à calculer à partir des variables présentées en direct les fonctions :

$$f = \bar{x}_0\, x_1\, \bar{x}_2\, \bar{x}_3\, x_4\, \bar{x}_5\, x_6\, x_7\, \bar{x}_8\, \bar{x}_9\, x_{10}\, x_{11}\, x_{12}\, x_{13}\, \bar{x}_{14}\, x_{15}$$

$$g = x_0 + \bar{x}_1 + \bar{x}_2 + x_3 + x_4 + \bar{x}_5 + x_6 + x_7 + \bar{x}_8 + x_9 + \bar{x}_{10} + x_{11} + \bar{x}_{12} + x_{13} + x_{14} + \bar{x}_{15}$$

La figure 8.1 montre le logigramme de la fonction f réalisée avec des NOR, NAND et inverseurs. Des fonctions ET partielles sont calculées puis regroupées.

D'une manière analogue, la figure 8.2 montre la réalisation de la fonction g, OU des seize variables x_0 à x_{15}, directes et inverses, à partir du calcul de OU partiel.

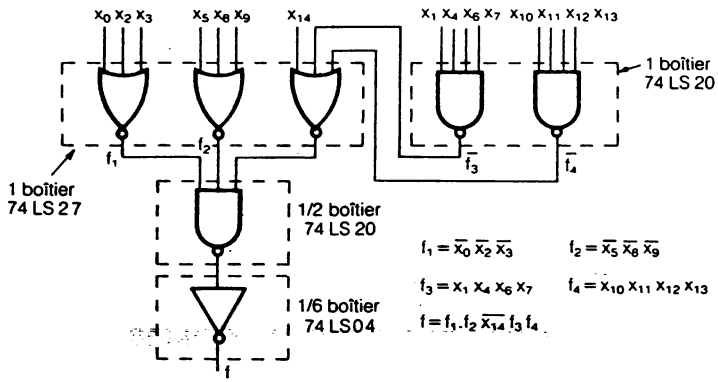


Fig. 8.1. — *Extension d'une fonction ET.*

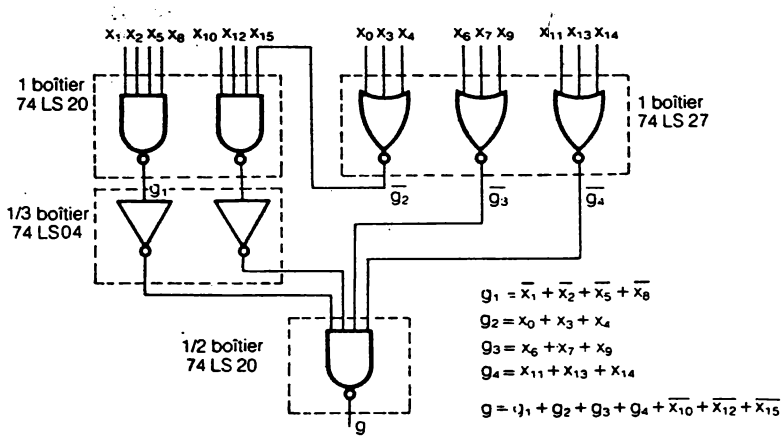


Fig. 8.2. — *Extension d'une fonction OU.*

3. EXTENSION DE LA FONCTION DE DÉCODAGE

On souhaite réaliser un décodage 1 parmi 32. On peut considérer que les 32 sorties se répartissent en quatre groupes de huit. Une sortie S_j, qui dépend de cinq variables x, y, z, t, u (x poids faible, u poids fort), se repère par une adresse de groupe g (1 parmi 4) et une adresse interne i (1 parmi 8) dans le groupe. On peut écrire:

$$S_j = (utzyx = j) = (ut = g).(zyx = i)$$

La fonction S_j peut ainsi s'interpréter comme un double décodage: 4 décodages 1 parmi 8 (xyz) validés chacun par une sortie de décodage 1 parmi 4 (ut). La figure 8.3 donne le schéma symbolique correspondant.

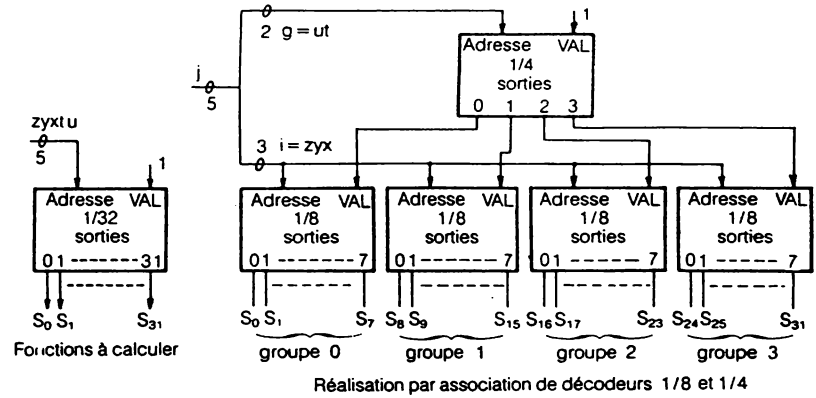


Fig. 8.3. — *Schéma symbolique d'un décodage 1 parmi 32.*

La figure 8.4 montre la transposition du schéma symbolique avec des décodeurs 74 LS 138.

On peut obtenir un schéma plus simple sans décodge de ut par un boîtier particulier, en utilisant les entrées de validation des boîtiers 74 LS 138. En effet, les sorties du groupe 0 s'écrivent:

$$S_j = (ut = 0).(zyx = i) = \overline{u}\,\overline{t}.(zyx = i)$$

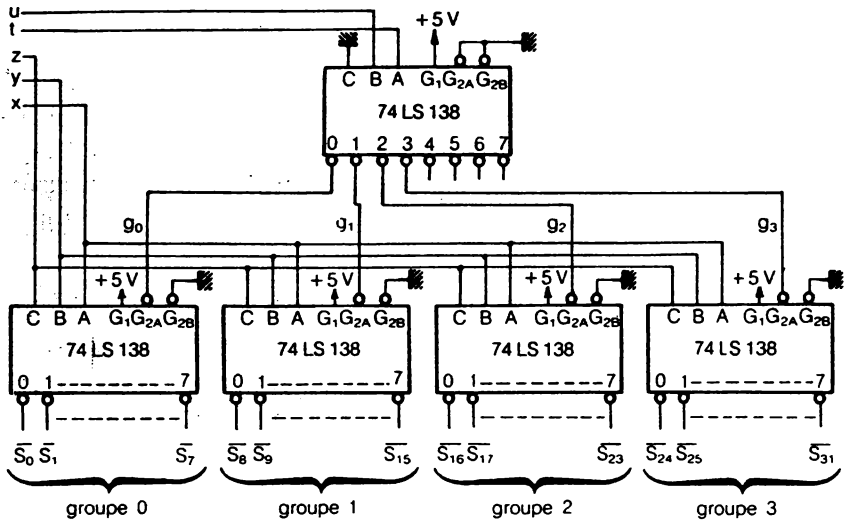


Fig. 8.4. — *Décodage 1 parmi 32 (1re version).*

ce qui s'interprète comme un décodage 1 parmi 8 de zyx validé par $\overline{u}\,\overline{t}$. Pour le réaliser, on applique ut sur les deux entrées inverses du décodeur (G_{2A} et G_{2B}) et xyz sur ABC. Pour le groupe 1, on a :

$$S_j = (ut = 1).(zyx = \hat{\imath}) = \overline{ut}.(zyx = \hat{\imath})$$

ce qui donne u sur G_{2A}, t sur G_1 et xyz sur ABC, et ainsi de suite pour les deux autres groupes. On obtient le schéma de la figure 8.5.

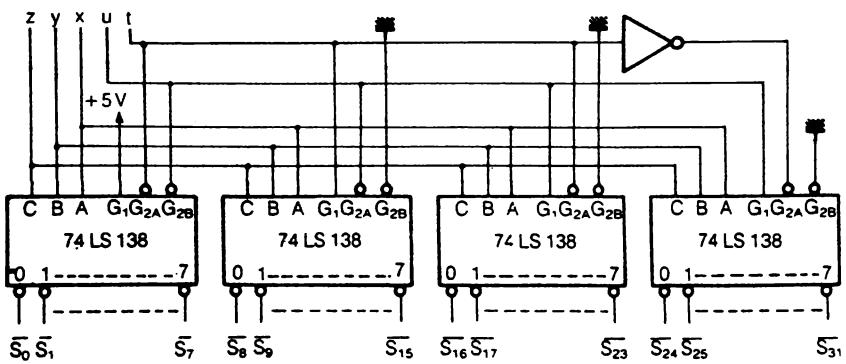


Fig. 8.5. — *Décodage 1 parmi 32 (2e version).*

4. EXTENSION DE LA FONCTION MULTIPLEXAGE

On souhaite réaliser un multiplexage 32 vers 1. On peut considérer que les 32 entrées se répartissent en quatre groupes de huit. La sortie S, qui dépend de cinq variables d'adresse $xyztu$ (x poids faible, u poids fort) et des 32 entrées E_0 à E_{31} s'écrit, si l'on considère qu'une entrée se repère par une adresse de groupe g (1 parmi 4) et une adresse i (1 parmi 8) interne au groupe.

$$S = \sum_{j=0}^{31} (utzyx = j).E_j$$

$$S = \sum_{g=0}^{3} \sum_{i=0}^{7} (ut = g).(zyx = i).E_{gi}$$

$$S = \underbrace{\sum_{g=0}^{3} (ut = g) \underbrace{(\sum_{i=0}^{7} (zyx = i).E_{gi})}_{\text{multiplexage 8 vers 1}}}_{\text{multiplexage 4 vers 1}}$$

La dernière formule s'interprète comme un multiplexage 4 vers 1 de 4 données provenant de 4 multiplexages 8 vers 1. La figure 8.6 donne le schéma du multiplexage 32 vers 1 utilisant des circuits 74 LS 151.

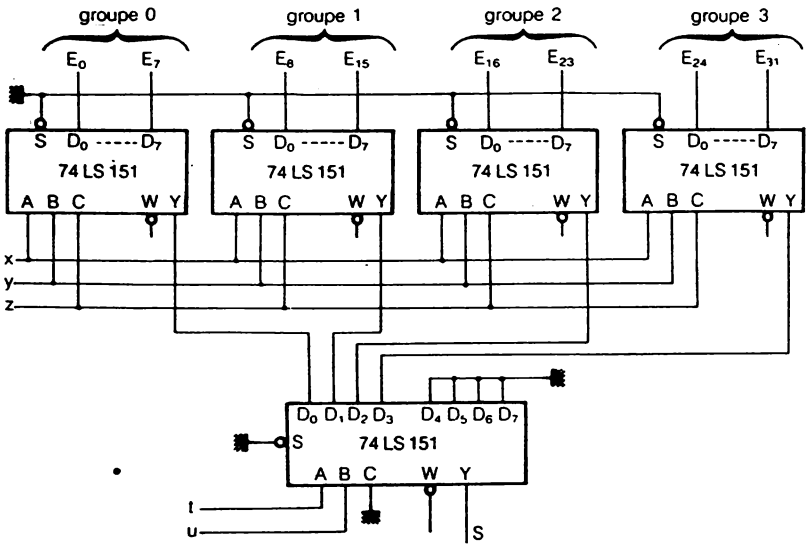


Fig. 8.6. — *Multiplexage 32 vers 1.*

5. EXTENSION DE LA FONCTION ENCODAGE

On souhaite déterminer l'adresse de l'entrée active la plus prioritaire dans un ensemble de 32 entrées E_0 à E_{31}, E_{31} étant la plus prioritaire. Cette adresse est constituée de 5 bits *xyztu* (*x* poids faible et *u* poids fort). On peut diviser cette adresse globale en deux parties: une adresse de groupe *g*, exprimée par *ut* (quatre possibilités) et une adresse interne au groupe *i*, exprimée par *xyz* (huit possibilités). Pour déterminer *g* et *i*, on place sur chacun des quatre groupes deux circuits: l'un qui indique la présence d'au moins une entrée active dans le groupe (fonction *p*), l'autre qui donne l'adresse *g* (sur trois éléments binaires) de l'entrée active de rang le plus élevé dans le groupe (fonction encodage).

On dispose donc de quatre adresses de groupes g_0, g_1, g_2, g_3 de quatre fonctions de présence d'entrée active, p_0, p_1, p_2, p_3 à partir desquelles on calcule:

— Une fonction adresse valide AV qui indique qu'il existe au moins une entrée active parmi les 32 entrées.

– L'adresse de groupe $g(tu)$.

– L'adresse interne $i(xyz)$.

La fonction AV se calcule simplement. C'est la réunion des quatre fonctions de présence :

$$AV = p_0 + p_1 + p_2 + p_3.$$

L'adresse g est égale au plus élevé des rangs des groupes ayant au moins une entrée active. Les groupes ayant au moins une entrée active se caractérisent par la fonction p. La fonction "le rang le plus élevé" est une fonction d'encodage que l'on applique à p_3, p_2, p_1, p_0, la fonction p_3 étant la plus prioritaire.

g = Encodage (p_3, p_2, p_1, p_0)

L'adresse i est égale soit à :

– g_3 si $p_3 = 1$

– ou à g_2 si $p_3 = 0$, $p_2 = 1$

– ou à g_1 si $p_3 = p_2 = 0$, $p_1 = 1$

– ou à g_0 si $p_3 = p_2 = p_1 = 0$, $p_0 = 1$

– d'où $i = g_3 p_3 + g_2 \overline{p}_3 p_2 + g_1 \overline{p}_3 \overline{p}_2 p_1 + g_0 \overline{p}_3 \overline{p}_2 \overline{p}_1 p_0$.

La figure 8.7 montre une réalisation de l'encodage à 32 entrées avec des encodeurs 74 LS 148. Cet encodeur a :

– une entrée de validation *EI* (Enable Input),

– huit entrées d'information,

– trois sorties d'adresse,

– une sortie *GS* qui indique la présence d'une entrée active,

– une sortie *EO* (Enable Output) qui sert à valider l'encodeur de poids inférieur lorsqu'il n'y a aucune entrée active.

Les sorties *GS* des 4 encodeurs associés aux groupes d'entrée donnent les quatre fonctions p_3, p_2, p_1, p_0 (en inverse). Leur encodage dans le cinquième encodeur fournit l'adresse de groupe t, u sur les sorties A, B et la sortie *GS* de ce cinquième encodeur donne la fonction adresse valide AV.

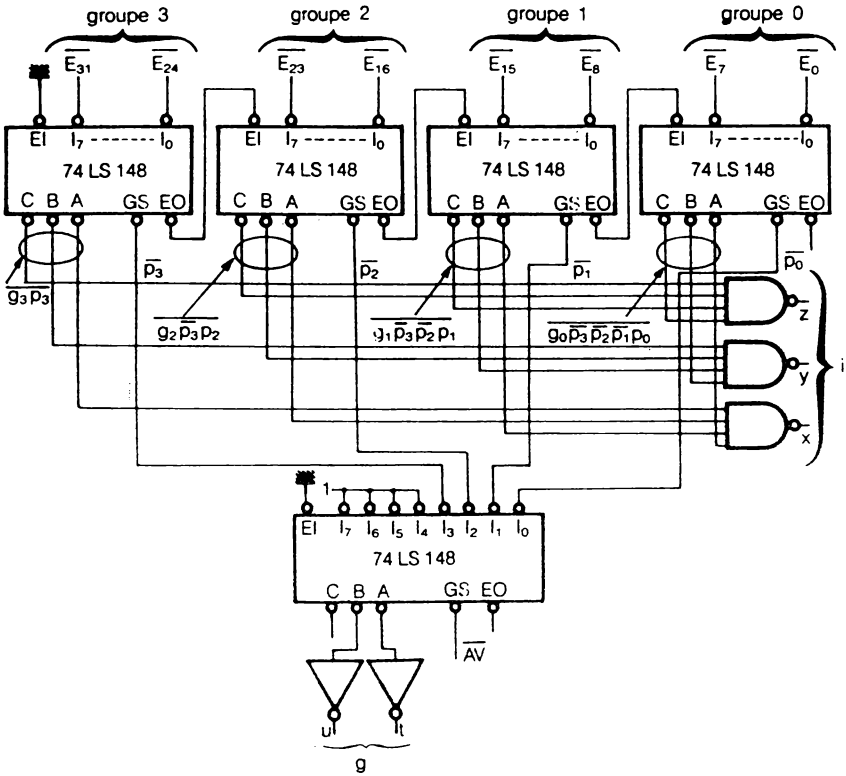


Fig. 8.7. — *Encodage à 32 entrées.*

La formule donnant l'adresse interne est obtenue par les trois portes NAND qui regroupent (en fonction OU) les quatre adresses internes. Les produits g_3p_3, $g_2\overline{p_3}p_2$, etc... sont réalisés à l'intérieur du boîtier grâce aux entrées *EI* et aux sorties *EO*.

6. EXTENSION DE LA FONCTION DE COMPARAISON ARITHMÉTIQUE

Le circuit 74 LS 85 compare des nombres de 4 eb ($A > B$, $A < B$, $A = B$). Il a été conçu pour être mis en cascade afin de comparer des nombres de grande taille. Les sorties $A > B$, $A < B$, $A = B$ du comparateur de poids *i* sont connectées aux entrées A > B, A < B, A = B homologues du

comparateur de poids supérieur. Le lecteur se reportera au chapitre 7 pour la théorie du comparateur et l'association de plusieurs comparateurs. La figure 8.8 montre une comparaison arithmétique sur 16 eb.

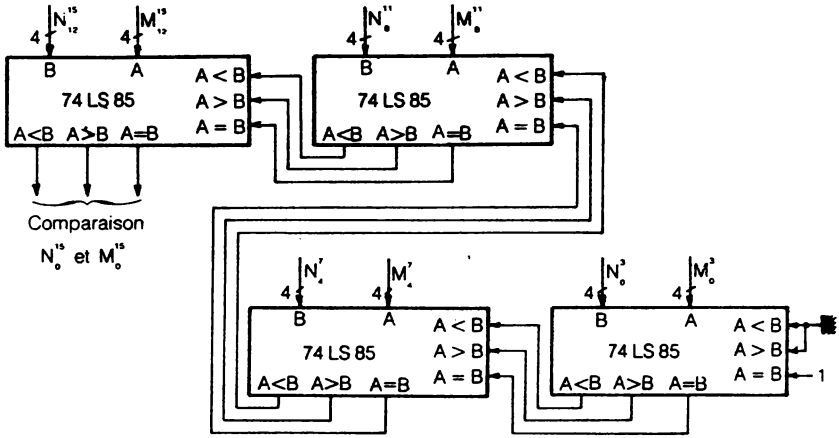
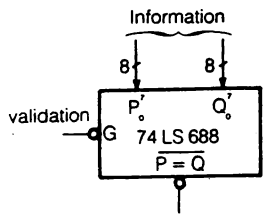


Fig. 8.8. — *Comparaison de deux nombres M et N de 16 eb.*

7. EXTENSION DE LA FONCTION ÉGALITÉ

Réalisons la fonction égalité des deux nombres M et N à 32 eb chacun. On utilise pour cela des comparateurs octaux 74 LS 688 (fig. 8.9). On peut faire une mise en cascade en série comme le montre la figure 8.10.



P'_0 Q'_0	G	$\overline{P = Q}$
X X	1	1
$P'_0 \neq Q'_0$	0	1
$P'_0 = Q'_0$	0	0

$\overline{(P = Q)} = \overline{G}.\overline{(P'_0 = Q'_0)}$

Fig. 8.9. — *Comparateur 74 LS 688.*

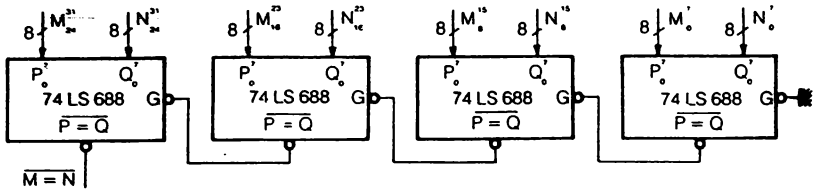


Fig. 8.10. — *Comparaison série.*

Calculons la sortie $(\overline{P=Q})$ de l'étage qui reçoit N_{24}^{31} et M_{24}^{31}

$$(\overline{P=Q}) = \overline{G_{24}^{31}}\,(M_{24}^{31} = N_{24}^{31}) \text{ or } \overline{G_{24}^{31}} = (\overline{P=Q})_{16}^{23} = \overline{G_{16}^{23}}\,(P_{16}^{23} = Q_{16}^{23})$$

$$(\overline{P=Q}) = \overline{G_{16}^{23}}\,(M_{24}^{1} = N_{24}^{31})\,(M_{16}^{23} = N_{16}^{23}) \text{ et ainsi de suite en descendant}$$

jusqu'à M_0^7 et N_0^7.

$$(\overline{P=Q}) = \overline{G_0^7}\,(M_{24}^{31} = N_{24}^{31})\,(M_{16}^{23} = N_{16}^{23})\,(M_8^{15} = N_8^{15})\,(M_0^7 = N_0^7)$$

Or $G_0^7 = 0$. Le produit des égalités est égal à l'égalité de l'ensemble des éléments binaires d'où $(\overline{P=Q}) = (\overline{M_0^{31} = N_0^{31}}) = (\overline{M=N})$.

Le temps de stabilisation de la sortie $M = N$ est égal à la somme du temps nécessaire à la comparaison de M_0^7 à N_0^7 pour afficher $\overline{P=Q}$ dans le comparateur de poids faible et trois fois le temps de validation de l'entrée G à la sortie $\overline{P=Q}$ pour les trois comparateurs suivants soit en prenant les valeurs maximales :

$23 + 3 \times 20 = 83$ *ns.*

Le schéma de la figure 8.11 donne un temps de calcul plus faible en utilisant un cinquième comparateur. Si M et N sont égaux, les sorties $\overline{P=Q}$ des quatre premiers comparateurs donnent 0 de sorte que le cinquième comparateur reçoit $Q_0^7 = P_0^7 = 0$. Sa sortie $P = Q$ indique l'égalité. Si M et N sont inégaux, l'une au moins des sorties $\overline{P=Q}$ des quatre premiers comparateurs donne 1, de sorte que le cinquième comparateur reçoit $Q_0^7 \neq 0$ et $P_0^7 = 0$. Sa sortie $\overline{P=Q}$ indique l'inégalité.

Le temps d'obtention de la fonction $\overline{M=N}$ est égal à deux fois le retard des entrées P, Q vers la sortie $\overline{P=Q}$ soit $2 \times 23 = 46$ *ns.*

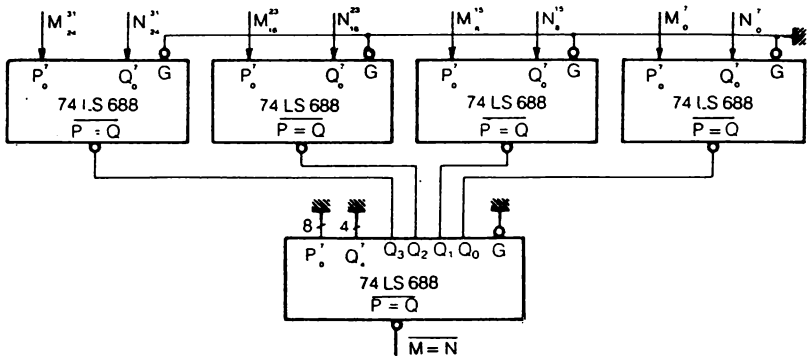


Fig. 8.11. — *Comparaison en parallèle.*

8. MISE EN CASCADE DE BLOC DE CALCULS

8.1. Propagation série des retenues

La mise en cascade d'additionneurs simples ne pose aucune difficulté. Il suffit de relier la retenue sortante d'un étage à la retenue entrante de l'étage de poids immédiatement supérieur. La figure 8.12 montre la mise en cascade d'additionneurs 74 LS 83 A et de blocs de calcul 74 LS 382.

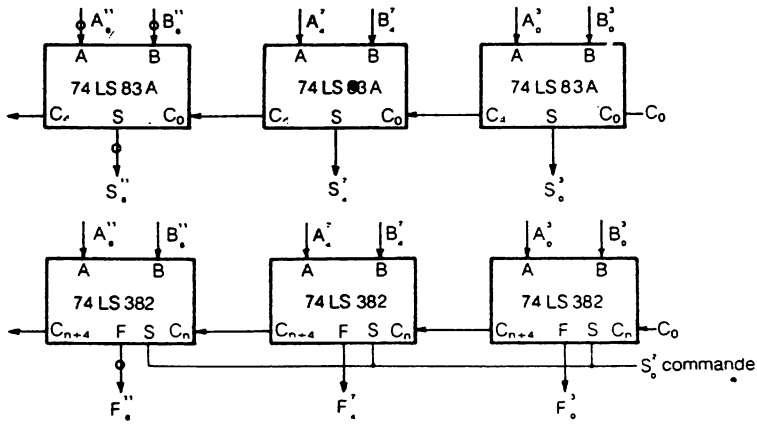


Fig. 8.12. — *Mise en cascade de blocs de calcul.*

Dans ces schémas, le calcul des derniers eb du résultat nécessite la propagation de la retenue entrante C_0 à travers les boîtiers de rang inférieur. La propagation de la retenue est parallèle à l'intérieur des boîtiers et série à l'extérieur.

8.2. Propagation entièrement parallèle des retenues

Pour aller plus vide il faut propager les retenues en parallèle à l'extérieur des boîtiers. Pour cela on utilise le même mécanisme à l'extérieur des boîtiers qu'à l'intérieur. Soit un bloc de calcul à quatre étage recevant une retenue entrante C_n et fournissant une retenue sortante C_{n+4}. D'après la formule vue au chapitre précédent:

$$C_l = G_l + P_l C_{l-1}$$

nous obtenons la retenue sortante:

$$C_{n+4} = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_n$$

On peut considérer qu'un tel bloc de calcul se comporte comme un seul étage ayant une fonction de génération G et une fonction de propagation P telle que:

$$C_{n+4} = G + P C_n$$

avec:

$$G = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1$$

$$P = P_4 P_3 P_2 P_1.$$

Pour connecter en parallèle les retenues de ces blocs de calculs il suffit que chacun d'eux fournisse les fonctions P et G et qu'un circuit extérieur recevant tous les P et G des blocs de calcul et la retenue initiale élabore les retenues entrantes dans chacun des blocs de calcul. La figure 8.13 montre le

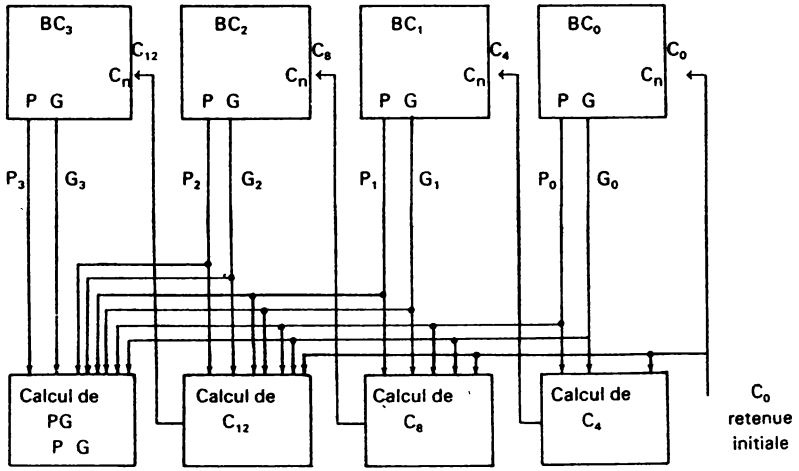


Fig. 8.13. — *Principe de la propagation parallèle des retenues à l'extérieur des blocs de calcul.*

principe. A partir de C_0 on calcule C_4, C_8, C_{12} et deux fonctions P G pour étendre éventuellement la connexion des blocs de calculs au-delà :

$$C_4 = G_0 + P_0 C_0$$

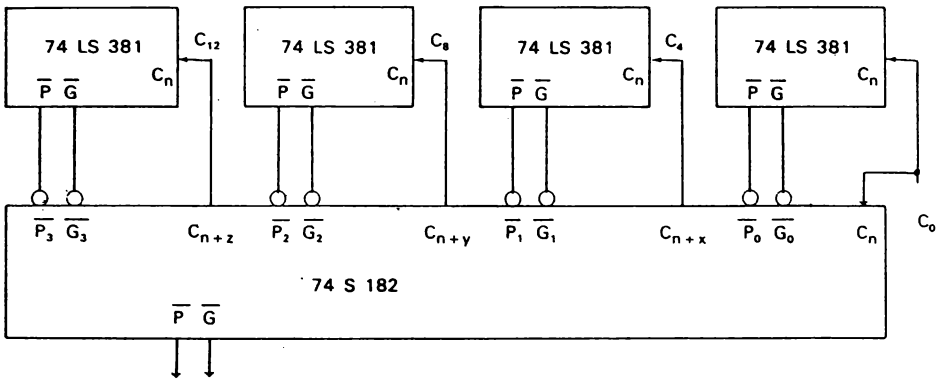
$$C_8 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_{12} = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

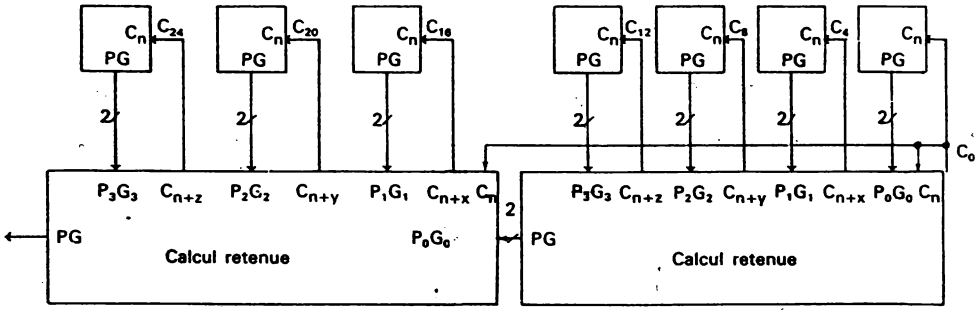
$$P = P_3 P_2 P_1 P_0$$

$$G = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0.$$

La figure 8.14 montre une réalisation pratique avec des blocs de calcul 74 LS 381 et un circuit de calcul de retenue anticipée 74 S 182.



bloc de calcul à 16 bits



Boc de calcul à 28 bits (principe).

Fig. 8.14. — *Connexion des retenues en parallèle.*

Comparons les performances des montages à propagation série et parallèle :

1. temps d'addition sur 16 étages avec 74 LS 382 :

$$t = t_{calcul\ retenue\ (382)} + t_{addition\ (382)}$$

$t = 3 \times 21 + 27 = 90\ ns$

2. temps d'addition sur 28 étages avec 74 LS 382 :

$t = 6 \times 21 + 27 = 153\ ns$

3. temps d'addition sur 16 étages avec 74 LS 381 :

$$t = t_{calcul\ PG(381)} + t_{calcul\ retenue\ (182)} + t_{addition\ (381)}$$

$t = 33 + 7 + 27 = 67\ ns$

4. temps d'addition sur 28 étages avec 74 LS 381 :

$$t = t_{calcul\ PG(381)} + t_{calcul\ PG(182)} + t_{calcul\ retenue\ (182)} + t_{addition\ (381)}$$

$t = 33 + 11 + 7 + 27 = 78\ ns$

On voit que le gain apporté par le circuit de calcul de retenue anticipée est notable pour 28 bits (près de la moitié).

Chapitre 9

CONCEPTION DES SYSTÈMES COMBINATOIRES

1. MÉTHODE DE CONCEPTION

Le but de la conception est d'obtenir un schéma à partir des spécifications du système. La figure 9.1 propose les étapes de la synthèse. Comme on le constate, il n'y a pas qu'une seule façon d'aboutir à un schéma.

Les spécifications constituent le point de départ. Très souvent le concepteur doit les compléter car tous les cas de figure ne sont pas toujours envisagés.

L'étape suivante, l'analyse, détermine les types de fonctions constituant le système. Trois cas peuvent se présenter :

- l'analyse met en évidence des fonctions booléennes connues ou des extensions de fonctions booléennes connues. On utilise alors directement les circuits MSI existants : décodeur, multiplexeur, encodeur prioritaire, etc. Le schéma s'obtient immédiatement (voir chap. 8).
- l'analyse met en évidence un système simple, c'est-à-dire ayant peu d'entrées et peu de sorties, mais dans lequel n'apparaissent pas les fonctions boolénnes connues, comme dans le cas précédent. On établit alors la table de vérité du système qui donne pour toutes les combinaisons des

entrées la valeur des fonctions de sortie. On choisit ensuite des composants avec le souci d'obtenir un schéma final optimal, c'est-à-dire comprenant le plus souvent le nombre minimum de boîtiers. Quatre possibilités s'offrent alors, que le concepteur peut mettre en œuvre séparémment ou conjointement :

- utiliser des portes ET, OU, NAND, NOR, ...c'est la méthode la plus courante. On calcule et l'on simplifie les fonctions à partir de la table de vérité. Les formules obtenues sont ensuite transposées avec des portes pour donner le schéma.

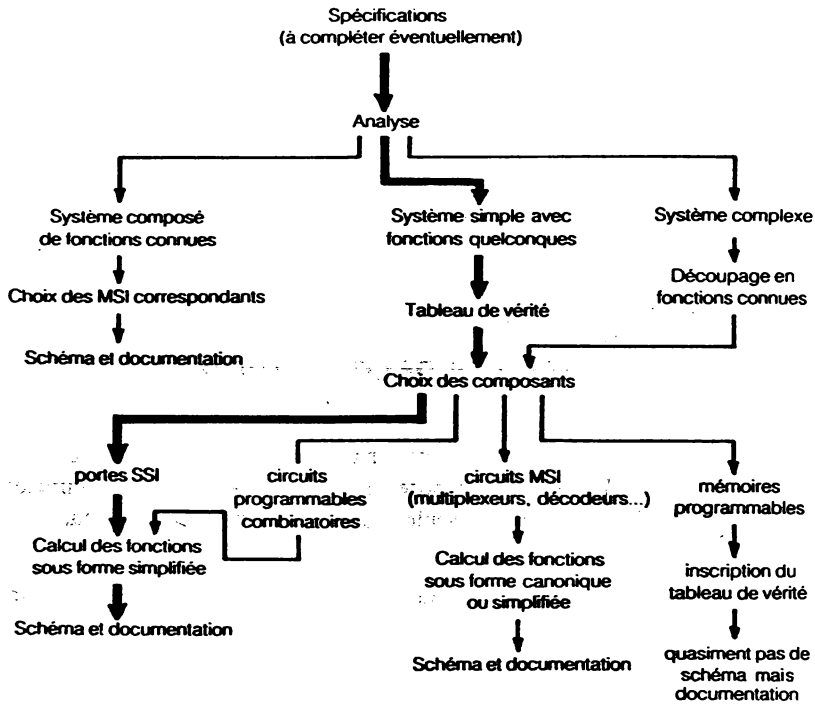


Fig. 9.1. — *Différentes méthodes de synthèse d'un système combinatoire.*

- utiliser des circuits programmables combinatoires. Ces circuits sont constitués de réseaux très denses de portes ET-OU. Ils sont étudiés au chapitre 15. On procède comme pour les portes ET, OU, ... vues plus haut.

- utiliser des circuits MSI (multiplexeur, décodeur, ...) comme générateurs de fonctions booléennes. Dans certains cas on pourra utiliser les fonctions sous forme canonique, sans les simplifier, dans d'autres cas une simplification des fonctions pourra conduire à un montage plus simple.
- utiliser des mémoires programmables PROM en générateur de fonctions booléennes. Ces mémoires sont étudiées aux chapitre 15. Il n'y a aucun calcul booléen ; la table de vérité est directement inscrite dans la mémoire. Le schéma se réduit à la simple connexion des variables d'entrée sur la mémoire.

— l'analyse met en évidence un système complexe, c'est-à-dire ayant beaucoup d'entrées, beaucoup de sorties. L'étude directe par un tableau de vérité, comme dans le cas précédent n'est pratiquement plus possible à cause du grand nombre de lignes que comporterait la table de vérité : n variables d'entrées donnent un tableau à 2^n lignes. Dans ce cas on essaie de décomposer le système global en plusieurs sous-systèmes combinatoires dans lesquels apparaissent :

- soit des sous-systèmes simples, calculables à partir d'un tableau de vérité,
- soit des fonctions connues, disponibles sous forme de boîtier : multiplexeur, décodeur, additionneur (fig. 9.2). Pour réaliser cette découpe, on ne dispose pas de méthode générale, et le concepteur s'appuie sur son expérience, sa connaissance des fonctions combinatoires MSI et son intuition. Le découpage en fonctions connues effectué, le concepteur choisit alors les composants et se retrouve dans la même filière que pour un système simple.

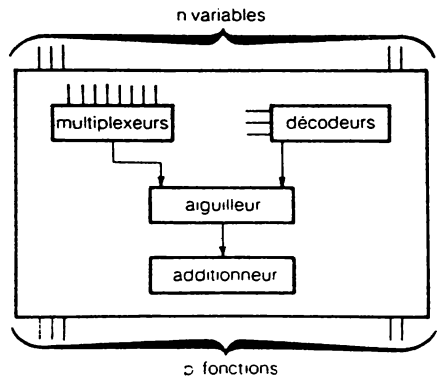


Fig. 9.2. — *Systèmes combinatoires décomposés en fonctions connues.*

Dans la figure 9.1, le lecteur pourra remarquer un trajet avec des flèches en trait gras. Il correspond à l'exposé généralement proposé pour la synthèse des systèmes combinatoires mais qui ne recouvre en fait qu'une faible partie des cas rencontrés dans la pratique.

Pour illustrer les différentes méthodes proposées nous traiterons successivement plusieurs cas.

2. ÉTUDE D'UN SYSTÈME COMPORTANT DES FONCTIONS CONNUES

On souhaite visualiser l'état de 64 dispositifs mécaniques pouvant être ouverts ou fermés. Un contact de relais représente cet état. La figure 9.3 donne une solution avec 64 lampes.

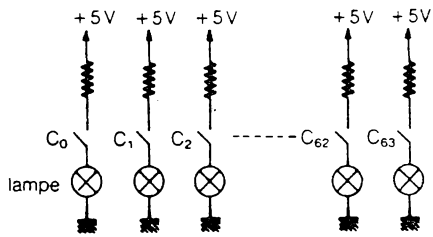


Fig. 9.3. — *Visualisation de 64 états.*

Or il s'avère (dans le problème posé) que l'utilisateur n'a pas besoin de connaître simultanément l'ensemble des états, mais celui d'un seul qu'il choisit. La visualisation porte alors sur l'état sélectionné. L'analyse fait ainsi apparaître une fonction connue : le multiplexage. La figure 9.4 donne le schéma correspondant : il s'agit d'un multiplexage de grande capacité étudié au chapitre 8.

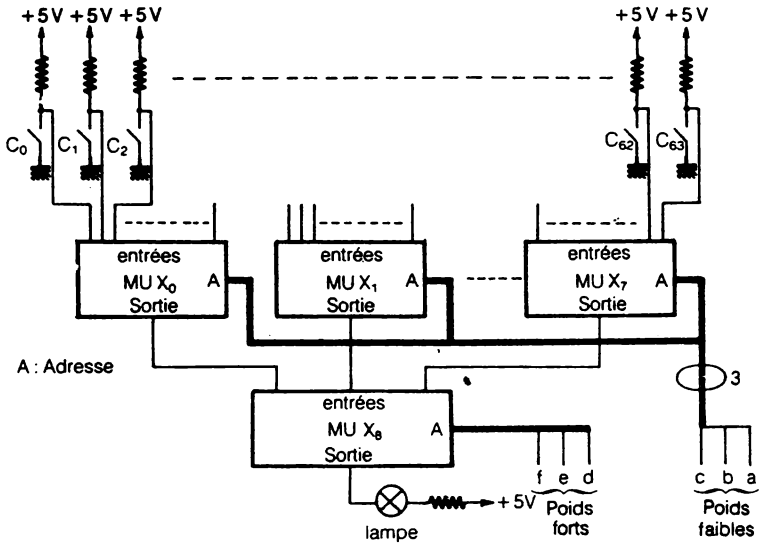


Fig. 9.4. — *Visualisation de l'état d'un organe parmi 64.*

3. ÉTUDE D'UN SYSTÈME SIMPLE

3.1. Spécification d'un distributeur de boisson

Un appareil comporte trois cuves contenant de l'eau, du cassis et de la menthe. Trois boutons *e*, *m*, *c*, commandant des électrovannes *E*, *M*, *C*, permettent d'obtenir de l'eau pure, de la menthe à l'eau ou du cassis à l'eau. Une pièce *p*, doit être introduite, sauf pour l'eau pure, qui est gratuite. En cas de fausse manœuvre (demande incorrecte) la pièce est rendue au bout d'une temporisation *T*. Une fonction *R* (Restitution de la pièce) est calculée à cette occasion. Pour simplifier on ne tient pas compte de la temporisation *T*.

3.2. Analyse

L'énoncé du problème ne suggère pas de fonction combinatoire courante. Par contre, il n'y a que quatre fonctions de sortie *E*, *M*, *C*, *R* ne

dépendant que des quatre variables d'entrée e, m, c, p. Il s'agit donc d'un système combinatoire simple, analysable avec un tableau de vérité.

3.3. Table de vérité

3.3.1. Établissement du tableau

Le tableau de vérité de la figure 9.5 donne l'état de chaque fonction E, M, C, R pour les 16 combinaisons des quatre variables e, m, c, p Pour chacun des 16 cas, nous déterminons les électrovannes à commander et la fonction de restitution de la pièce.

- *Combinaison 0*: aucune commande n'est effectuée $e = m = c = p = 0$. Aucune action n'est donc à entreprendre, donc $E = M = C = 0$ et $R = 0$.
- *Combinaison 1*: une pièce est introduite, mais suivie d'aucun choix. Au bout de la temporisation déclenchée par l'introduction de la pièce, la machine restitue celle-ci, mais ne commande pas les électro-vannes donc $E = M = C = 0$ et $R = 1$.
- *Combinaison 2*: action de l'opérateur sur le bouton cassis mais aucune pièce n'est introduite dans le temps imparti. Aucune action n'est à entreprendre $E = M = C = R = 0$.
- *Combinaison 3*: une pièce est introduite et l'opérateur agit sur le bouton cassis. D'où les commandes:

  vanne cassis $C = 1$
  vanne eau $E = 1$ (sinon l'opérateur risque d'avoir beaucoup plus soif qu'avant).

  Il n'y a aucune raison de lui donner de la menthe ($M = 0$) et de lui restituer la pièce ($R = 0$) puisque la prestation a été rendue.
- *Combinaison 4*: action sur le bouton menthe seul et pas d'introduction de pièce donc $E = M = C = R = 0$.
- *Combinaison 5*: l'opérateur réclame de la menthe et paie. Grand seigneur nous lui donnons une menthe à l'eau $E = M = 1$ et gardons la pièce $R = 0$, tandis que $C = 0$.
- *Combinaison 6*: l'opérateur agit sur les boutons menthe et cassis et n'introduit pas de pièce. Les coktails gratuits n'étant pas prévus, on a $E = M = C = R = 0$.

Combinaisons des variables	Variables				Fonctions			
	e	m	c	p	E	M	C	R
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	0	0
3	0	0	1	1	1	0	1	0
4	0	1	0	0	0	0	0	0
5	0	1	0	1	1	1	0	0
6	0	1	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1
8	1	0	0	0	1	0	0	0
9	1	0	0	1	1	0	0	1
10	1	0	1	0	0	0	0	0
11	1	0	1	1	1	0	1	0
12	1	1	0	0	0	0	0	0
13	1	1	0	1	1	1	0	0
14	1	1	1	0	0	0	0	0
15	1	1	1	1	0	0	0	1

Fig. 9.5. — *Table de vérité des fonctions E, M, C et R.*

— *Combinaison 7*: l'opérateur, cette fois-ci, plus honnête, agit sur les commandes menthe et cassis et introduit une pièce. Malheureusement les coktails, même payants, ne sont pas fournis. $E = M = C = 0$ et $R = 1$.

— *Combinaison 8*: l'opérateur agit uniquement sur le bouton eau, l'eau étant gratuite cette seule action déclenche la commande de la vanne eau $E = 1$ et $M = C = R = 0$.

— *Combinaison 9*: l'opérateur trop honnête, ou mal informé, agit sur le bouton eau et introduit une pièce.

Nous lui délivrerons un verre d'eau $E = 1$ et nous restituerons la pièce $R = 1$ ($M = C = 0$).

— *Combinaison 10*: l'opérateur séduit par la combinaison précédente se demande si en agissant sur eau et cassis et sans introduire de pièce, satisfaction va lui être donnée. Malheureusement pour lui nous avons prévu le cas $E = M = C = R = 0$.

— *Combinaison 11*: notre opérateur qui a toujours soif et tient à son cassis à l'eau reprend le cycle des opérations précédentes. Action sur les boutons eau et cassis et introduction d'une pièce. Nous ne pouvons pas refuser d'étancher sa soit $E = C = 1$ et $M = R = 0$.

— *Combinaison 12*: encore un fraudeur qui veut de la menthe gratuite $E = M = C = R = 0$.

- *Combinaison 13*: une menthe payée $E = M = 1$ et $C = R = 0$.
- *Combinaison 14*: client hésitant qui n'a pas payé $E = M = C = R = 0$.
- *Combinaison 15*: client payant qui promène ses doigts partout et dont il est impossible de déterminer les préférences. Nous restituerons la pièce en espérant qu'il prendra connaissance du mode d'emploi de l'appareil $E = M = C = 0$ et $R = 1$.

3.4. Choix des composants

Pour illustrer la méthode donnée à la figure 9.1, nous réaliserons le schéma du distributeur de boisson successivement avec des portes SSI, un circuit programmable combinatoire, des circuits MSI et une mémoire programmable.

3.4.1. Réalisation avec des portes SSI

On doit dans ce cas obtenir des formes simplifiées des fonctions de sortie. On reporte dans des tableaux de Karnaugh les valeurs de E, M, C, R et l'on simplifie (fig. 9.6).

Nous obtenons finalement en remarquant que E est une fonction de M et de C, les expressions suivantes:

$$M = m\bar{c}p$$
$$C = \bar{m}cp$$
$$E = \bar{e}\bar{m}\bar{c} + m\bar{c}p + \bar{m}cp = \bar{e}\bar{m}\bar{c} + M + C$$
$$R = \bar{m}\bar{c}p + mcp$$

Les expressions obtenues contiennent sous une forme condensée l'ensemble des propositions définies lors des seize combinaisons. Nous pouvons "lire" les fonctions obtenues et vérifier leur vraisemblance.

$$M = m\bar{c}p$$

La vanne menthe est ouverte si l'opérateur agit sur la vanne menthe (m) et pas sur la vanne cassis ($\bar{c}$) et à introduit une pièce (p). La formule recouvre les combinaisons 5 et 13.

$$C = \bar{m}cp$$

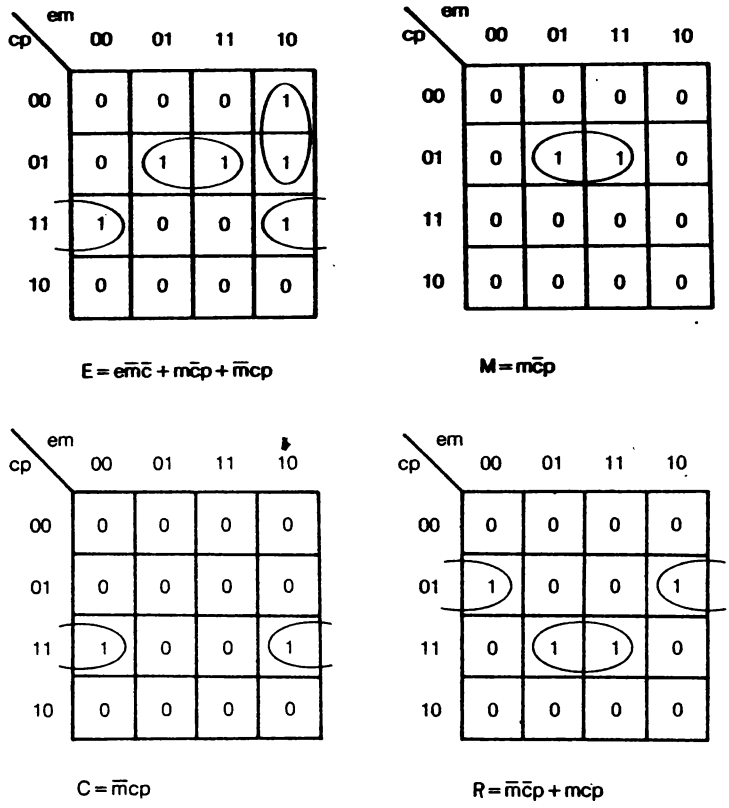


Fig. 9.6. — *Tableau de Karnaugh des fonctions E, M, C, R.*

La vanne cassis est ouverte si l'opérateur a demandé du cassis (c), pas de la menthe ($\bar{m}$) et a introduit une pièce (p). La formule recouvre les combinaisons 3 et 11.

$$R = (\bar{m}\,\bar{c} + mc)\ p$$

La machine rend la pièce si elle a été introduite "et" s'il y a eu une fausse manœuvre (menthe "et" cassis demandés simultanément) "ou" si il n'y a pas eu de demande de cassis "et" pas de menthe. Celle relation recouvre les propositions relatives aux combinaisons 1, 7, 9, 15. L'établir directement sans passer par l'analyse puis la synthèse précédente aurait été très difficile.

$$E = e\bar{m}\,\bar{c} + M + C$$

La vanne eau est ouverte si :

- l'opérateur agit sur le bouton eau (e) et n'agit pas sur le bouton menthe ($\bar{m}$) "et" sur le bouton cassis ($\bar{c}$),
- "ou" si la vanne menthe est commandée,
- "ou" si la vanne cassis est commandée.

Cette relation recouvre les propositions relatives aux combinaisons 3, 5, 8, 9, 11, 13. Il est évidemment impossible d'établir directement une telle formule.

La figure 9.7 donne le logigramme des quatre fonctions E, M, C, R avec des portes NAND et des inverseurs.

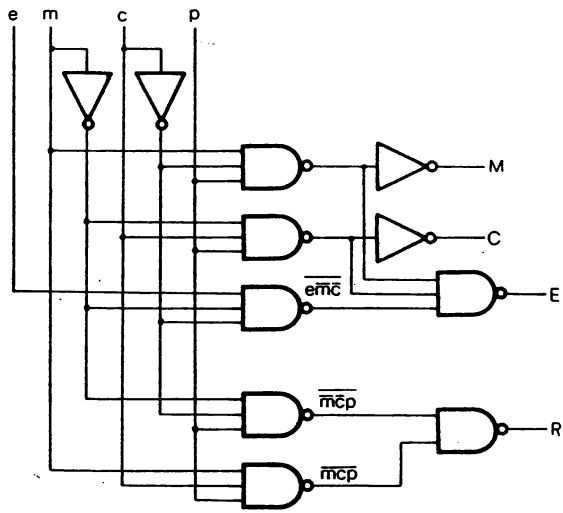


Fig. 9.7. — *Réalisation de E, M, C, R avec des portes simples.*

Le bilan en composants s'établit à :

6 portes NAND à 3 entrées	2,00 boîtiers	(3 portes par boîtier)
4 inverseurs	0,66 boîtier	(6 inverseurs par boîtier)
1 porte NAND à 2 entrées	0,25 boîtier	(4 portes par boîtier)
Total	3,91 boîtiers	

Nous comptons des fractions de boîtiers car dans un ensemble important les portes non utilisées pour le calcul de E, M, C, R serviraient à d'autres fonctions.

3.4.2. Utilisation de circuits programmables

Nous choisissons un circuit PAL de type 12H6. Il comprend 12 variables d'entrées e_1 à e_{12}, six fonctions OU de sortie S_1 à S_6. Deux OU regroupent quatre termes produits et quatre OU regroupent deux termes produits. La figure 9.8 montre la programmation de ce circuit pour obtenir les quatre fonctions E, M, C, R exprimées sous forme simplifiées en fonction de e, m, c, p comme dans le paragraphe précédent.

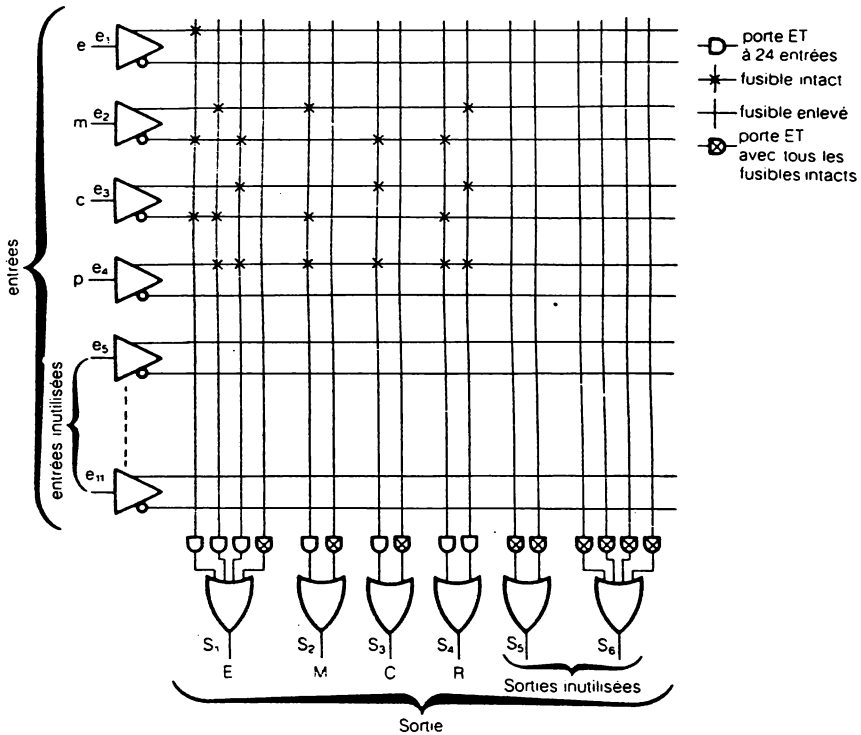


Fig. 9.8. — *Réalisation des fonctions E, M, C, R avec un PAL 12H6.*

La programmation s'effectue sans difficulté, par exemple :

la sortie S_1 calculant la fonction E donne:

$$S_1 = \bar{e}\bar{m}\bar{c} + m\bar{c}p + \bar{m}cp + \underbrace{e\bar{e}mm\bar{c}c...}_{0} = E$$

le bilan s'établit à 1 boîtier incomplètement utilisé. Nous pourrions encore calculer deux autres fonctions S_5 et S_6 de 12 variables.

3.4.3. Utilisation de circuits MSI

3.4.3.1. Multiplexeur générateur de fonctions booléennes

Le multiplexeur sert principalement à faire de l'aiguillage ou du multiplexage. Il peut être utilisé également en générateur de fonctions booléennes. Prenons par exemple le multiplexeur 74 LS 151 A (fig. 9.9).

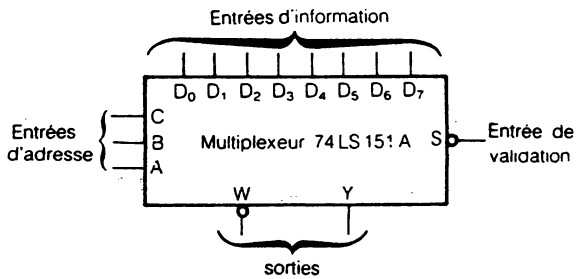


Fig. 9.9. — *Multiplexeur 74 LS 151 A.*

La sortie Y est égale à:

$$Y = \bar{S}.\sum_{i=0}^{7} (CBA = i).D_i$$

Écrivons maintenant la formule générale d'une fonction de quatre variables $x\ y\ z\ t$. Elle comprend au maximum seize monômes. Certains existent, d'autres pas, selon la valeur 0 ou 1 des coefficients α_i et β_i.

$$F = \alpha_0\,\bar{t}\bar{z}\bar{y}\bar{x} + \alpha_1\,\bar{t}\bar{z}\bar{y}x + ... \qquad \alpha_7\,\bar{t}zyx$$
$$+ \beta_0\,t\bar{z}\bar{y}\bar{x} + \beta_1\,t\bar{z}\bar{y}x + \qquad \beta_7\,tzyx$$
$$F = \bar{z}\bar{y}\bar{x}(\alpha_0\,\bar{t} + \beta_0\,t) + \bar{z}\bar{y}x(\alpha_1\,\bar{t} + \beta_1\,t) + ... zyx\,(\alpha_7\,\bar{t} + \beta_7\,t)$$

$$F = \sum_{i=0}^{7} (zyx = i).(\alpha_i \bar{t} + \beta_i t)$$

On remarque donc qu'une fonction quelconque F de quatre variables s'écrit d'une manière très voisine de la sortie d'un multiplexeur à trois entrées d'adresse.

$$\begin{cases} F = \sum_{i=0}^{7} (zyx = i).(\alpha_i \bar{t} + \beta_i t) \\ Y = \bar{S}.\sum_{i=0}^{7} (CBA = i).D_i \end{cases}$$

Pour réaliser une fonction F quelconque de quatre variables $xyzt$ avec le multiplexeur à trois entrées d'adresse CBA, une entrée de validation S, huit entrées d'information Di et une sortie Y, il faut identifier les deux formules de la manière suivante :

- Y donne la sortie F
- $\bar{S} = 0$
- xyz sont appliquées sur les entrées ABC respectivement.
- $\alpha_i \bar{t} + \beta_i t$ est câblé sur l'entrée D_i comme suit :

– si le monôme $(zyx = i)$ n'existe pas, on a $\alpha_i = \beta_i = 0$. On câble $\alpha_i \bar{t} + \beta_i t = 0\,\bar{t} + 0\,t = 0$ sur D_i.

– si le monôme $(zyx = i).\bar{t}$ existe, on a $\alpha_i = 1$, $\beta_i = 0$. On câble $1\,\bar{t} + 0\,t = \bar{t}$ sur D_i.

– si le monôme $(zyx = i).t$ existe, on a $\alpha_i = 0$, $\beta_i = 1$. On câble $0\,\bar{t} + 1\,t = t$ sur D_i.

– si le monôme $(zyx = i)$ existe de manière indépendante de t ou $\bar{t}$, on a $\alpha_i = \beta_i = 1$. On câble $1\,\bar{t} + 1\,t = 1$ sur D_i.

D'une manière générale un multiplexeur à n entrées d'adresse peut engendrer une fonction quelconque de $n + 1$ variables. L'entrée de validation peut servir à introduire une autre variable intervenant en produit dans tous les monômes. La figure 9.10 donne un exemple de réalisation.

$$F = u(\underbrace{\bar{t}\bar{z}yx}_{D3} + \underbrace{tz\bar{y}\bar{x}}_{D4} + \underbrace{\bar{z}\bar{y}x}_{D1})$$

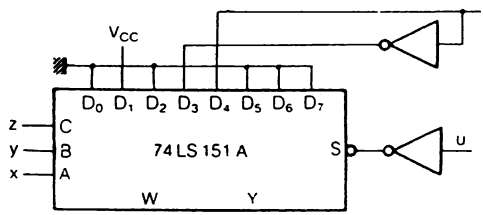


Fig. 9.10. — *Calcul d'une fonction avec un multiplexeur.*

3.4.3.2. Décodeur générateur de fonction booléenne

Le décodeur à n entrées d'adresse sert essentiellement à mettre une sortie parmi 2^n dans un état différent de toutes les autres. Il peut également fonctionner en générateur de fonctions booléennes. Prenons un décodeur 3 vers 8 du type 74 LS 138, et réunissons certaines des sorties Y_i, Y_j, Y_k par une porte NAND. La fonction sortant de la porte NAND est:

$$F = \overline{Y_i\, Y_j\, Y_k} = \overline{Y}_i + \overline{Y}_j + \overline{Y}_k$$

or une sortie m du décodeur vaut:

$$Y_m = G_1\, \overline{G}_{2A}\, \overline{G}_{2B}\, (CBA = m)$$

d'où

$$F = G_1\, \overline{G}_{2A}\, \overline{G}_{2B}\, ((CBA = i) + (CBA = j) + (CBA = k))$$

Les termes entre parenthèse représentent une somme de monômes, présents si la sortie Y_i a été câblée. On a donc une forme générale du type:

$$F = G_1\, \overline{G}_{2A}\, \overline{G}_{2B} \sum_{i=0}^{2^n} \alpha_i (CBA = i)$$

$\alpha_i = 1$ indiquant que le terme Y_i a été câblé sur le NAND. Cette forme correspond à celle d'une fonction booléenne.

$$F = \sum_{i=0}^{2^n} \alpha_i (zyx = i)$$

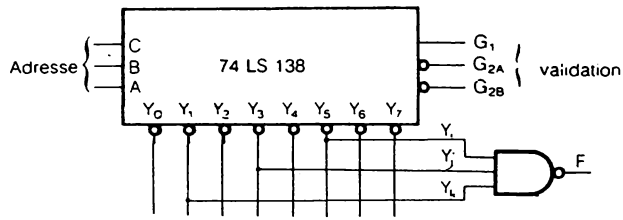


Fig. 9.11. — *Décodeur 74 LS 138 en générateur de fonction.*

Pour engendrer une ou plusieurs fonctions booléennes avec un décodeur et des portes NAND, on applique $G_1 = 1$, $G_{2A} = G_{2B} = 0$ et xyz sur les entrées ABC. On regroupe ensuite sur différentes portes NAND les sorties correspondant aux monômes.

Ainsi les fonctions $F_1 = \underbrace{\bar{z}\bar{y}\bar{x}}_{0} + \underbrace{z\bar{y}\bar{x}}_{6} + \underbrace{\bar{z}y\bar{x}}_{3}$ et $F_2 = \underbrace{z\bar{y}\bar{x}}_{5} + \underbrace{\bar{z}\bar{y}\bar{x}}_{0}$

se calculent en réunissant les termes 0, 3, 6 et 0, 5.

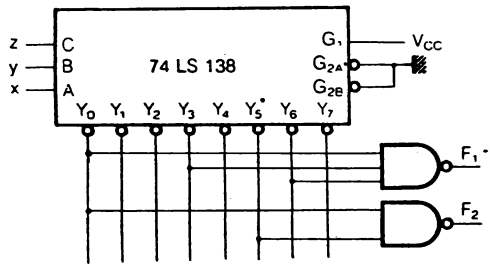


Fig. 9.12. — *Calcul de fonctions par un décodeur.*

Un décodeur à n entrées d'adresse permet de calculer plusieurs fonctions quelconques de n variables. Les entrées de validation peuvent servir à introduire d'autres variables en produit commun.

3.4.3.3. Calcul des fonctions E, M, C, R avec des multiplexeurs

Les formes canoniques des fonctions E, M, C, R, dépendant des 4 variables e, m, c, p s'obtiennent à partir du tableau de vérité de la figure 9.5.

$E = \Sigma\ (3, 5, 8, 9, 11, 13)$

$M = \Sigma\ (5, 13)$

$C = \Sigma\ (3, 11)$

$R = \Sigma\ (1, 7, 9, 15)$

Le calcul des fonctions sous forme canonique nécessite quatre multiplexeurs à trois entrées d'adresse. Si l'on utilise les formes simplifiées de E, M, C, R, on peut réduire le volume du matériel.

$M = pm\bar{c},\ C = p\bar{m}c,\ R = p\bar{m}\bar{c} + pmc$

$E = e\bar{m}\bar{c} + pm\bar{c} + \bar{p}mc$

On note que les trois premières fonctions M, C, R ont p en facteur commun. Il reste alors des termes à deux variables m et c que l'on peut réaliser avec un multiplexeur à une entrée d'adresse.

$$M = p(m\bar{c})\ \ C = p(\bar{m}c)\ \ R = p(\bar{m}\bar{c}) + (mc)$$
$$\downarrow\ \ \downarrow\ \ \downarrow\ \ \downarrow$$
$$0\ \ 1\ \ 0\ \ 1$$

Si l'on prend un multiplexeur quadruple 74 LS 157, il peut calculer les trois fonctions M, C, R. Rapprochons les formules de sortie du multiplexeur et les formules M, C, R.

$1Y = \bar{G}.(1A.\bar{S} + 1B.S)$ $\qquad M = p\,(m.\bar{c})$

$2Y = \bar{G}.(2A.\bar{S} + 2B.S)$ $\qquad C = p\,(\bar{m}.c)$

$3Y = \bar{G}.(3A.\bar{S} + 3B.S)$ $\qquad R = p\,(\bar{m}.\bar{c} + m.c)$

il vient $\bar{G} = p$ ou $G = \bar{p}$, $S = c$, $1A = m$, $1B = 0$, $2A = 0$, $2B = \bar{m}$, $3A = \bar{m}$, $3B = m$. On obtient le schéma de la figure 9.13.

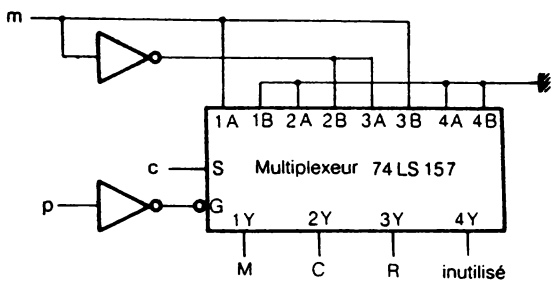


Fig. 9.13. — *Calcul de MCR avec un multiplexeur.*

Pour calculer E qui dépend des quatre variables e, m, c, p on peut prendre un multiplexeur à trois entrées d'adresse sur lesquelles on applique mcp. La figure 9.5 permet d'obtenir la formule canonique et d'en déduire le câblage.

$$E = \bar{e}(mcp = 3) + \bar{e}(mcp = 5) + e(mcp = 0) + e(mcp = 1) + e(mcp = 3) + e(mcp = 5)$$

$$E = (mcp = 3) + (mcp = 5) + e(mcp = 0) + e(mcp = 1)$$

Si l'on prend un multiplexeur 8 vers 1 de type 74 LS 151 A :

$$Y = \bar{S}.\sum_{i=0}^{7} (CBA = i)\, D_i, \text{ il vient}$$

$\bar{S} = 1$ soit $S = 0$, mcp appliqué à CBA respectivement, $D3 = D5 = 1$, $D0 = D1 = e$ et toutes les autres entrées sont à 0. On obtient le schéma de la figure 9.14.

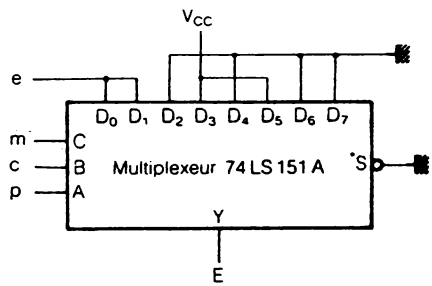


Fig. 9.14. – *Calcul de E.*

Une troisième façon enfin utilise deux multiplexeurs doubles 74 LS 153 à deux entrées d'adresse. Le premier calcule les fonctions M et C, et le deuxième les fonctions R et E. Rapprochons les formules de sortie des multiplexeurs et les formules à calculer.

$$1Y = \overline{1G}.\Sigma\ 1C_i(BA = i) \qquad M = p\underbrace{\bar{m}\bar{c}}_{2}$$

$$2Y = \overline{2G}.\Sigma\ 2C_i(BA = i) \qquad C = p\underbrace{\bar{m}c}_{1}$$

$$1Y = \overline{1G}.\Sigma\ 1C_i(BA = i) \qquad R = p\underbrace{\bar{m}\bar{c}}_{0} + p\underbrace{mc}_{3}$$

$$2Y = \overline{2G}.\Sigma\ 2C_i(BA = i) \qquad E = e\underbrace{\bar{m}\bar{c}}_{0} + p\underbrace{m\bar{c}}_{2} + p\underbrace{\bar{m}c}_{1}$$

Pour le premier multiplexeur, il vient $CBA = mc$, $1G = 2G = 0$, $1C_2 = p$, $2C_1 = p$ et les autres à 0. Pour le second multiplexeur, on a $1G = 2G = 0$, $1C_0 = p$, $1C_3 = p$, $2C_0 = e$, $2C_1 = p$, $2C_2 = p$. D'où le câblage de la figure 9.15.

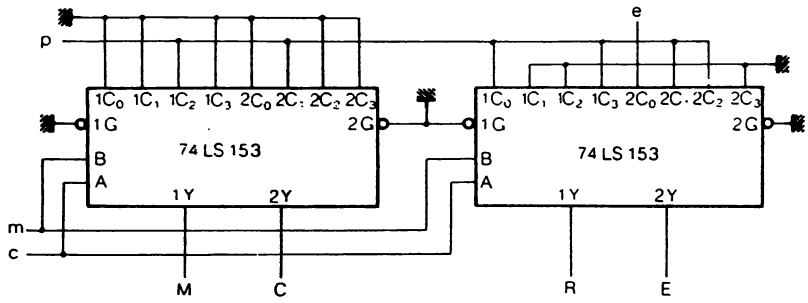


Fig. 9.15. — *Calcul de M, C, R, E.*

On remarque que le deuxième multiplexeur calcule une fonction E à quatre variables alors que la théorie indique qu'un multiplexeur à deux entrées d'adresse peut calculer une fonction quelconque de trois variables. Cette possibilité résulte de la forme particulière de E. Le monôme $\bar{m}\bar{c}$ n'est validé que par e et ne dépend pas de p.

Ce dernier schéma comprend deux boîtiers.

3.4.3.4. Calcul des fonctions E, M, C, R avec des décodeurs

Comme dans le cas précédent, avec les multiplexeurs il y a intérêt à travailler avec la forme réduite. On note que si l'on met p ou e en facteur dans les formules les termes restant ne dépendent plus que de m et c.

$$M = p\underbrace{(m\bar{c})}_{2},\quad C = p\underbrace{(\bar{m}c)}_{1}$$

$$E = e\underbrace{(\bar{m}\bar{c})}_{0} + p\underbrace{(m\bar{c})}_{2} + p\underbrace{(\bar{m}c)}_{1}$$

$$R = p\underbrace{(\bar{m}\bar{c})}_{0} + p\underbrace{(mc)}_{3}$$

On prend un double décodeur 74 LS 155, qui possède deux décodages de deux variables validées par des entrées différentes :

$$1Y_i = (AB = i)\ \overline{1G}\ 1C$$
$$2Y_i = (AB = i)\ \overline{2G}\ \overline{2C}$$

On place *mc* sur *BA* respectivement et l'on valide les sorties.

$1Y$ par $1G = 0$, $1C = p$ et les sorties $2Y$ par $2G = 0$, $\overline{2C} = \bar{e}$. On regroupe ensuite les monômes (fig. 9.16).

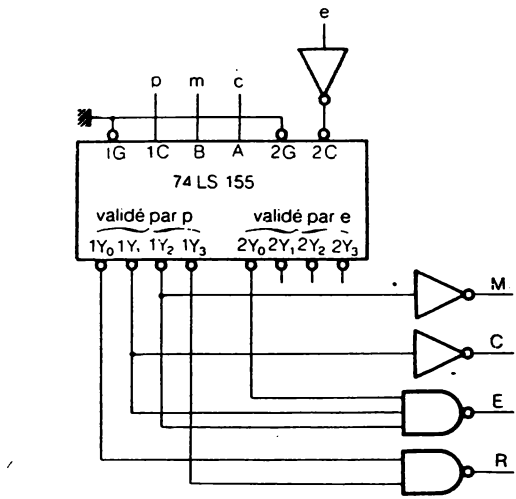


Fig. 9.16. — *Calcul des fonctions E, M, C, R par un décodeur.*

Le bilan s'établit à :

1 décodeur 74 LS 155	1
1 porte à 3 entrées	0,33
1 porte à 2 entrées	0,25
3 inverseurs	0,50
Total	2,08 boîtiers.

3.4.4. Utilisation de mémoire programmable

3.4.4.1. Mémoire PROM, générateur de fonctions

Soit une mémoire PROM ayant n eb d'adresse et des mots à p eb (fig. 9.17).

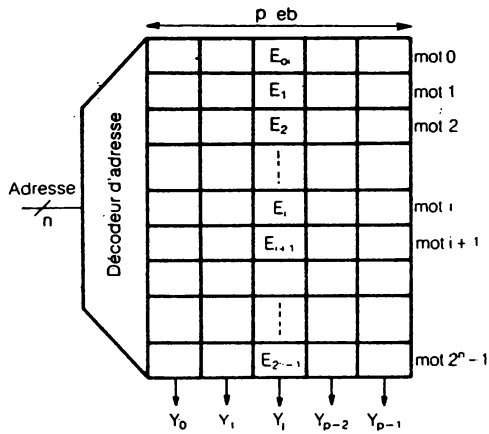


Fig. 9.17. — *Mémoire PROM.*

Si l'on examine l'état de la sortie Y_j de rang j, elle est égale :

- à l'état de l'eb j du mot 0,
- ou à l'état de l'eb j du mot 1,
- etc... ou à l'état de l'eb j du mot $2^n - 1$.

Finalement, la sortie Y_j s'exprime comme une somme :

$$Y_j = E_{oj}(Ad = 0) + E_{1j}(Ad = 1) + E_{2j}(Ad = 2) + \ldots$$
$$E_{2^n-1,j}(Ad = 2^n - 1)$$

$$Y_j = \sum_{i=0}^{2^n-1} E_{ij}(Ad = i)$$

Cette forme correspond à l'écriture canonique d'une fonction quelconque de n variables, les termes E_{ij} valant 1 ou 0 et déterminant les monômes de la fonction suivant qu'ils existent ($E_{ij} = 1$) ou non ($E_{ij} = 0$).

Une mémoire PROM ayant 2^n mots de p eb permet de calculer p fonctions quelconques de n variables. A titre d'exemple la figure 9.18 montre quatre fonctions de trois variables réalisées avec une PROM.

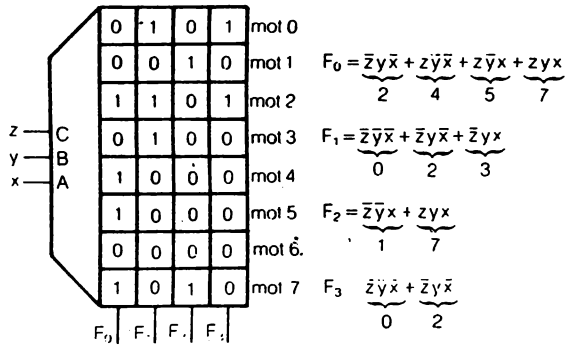


Fig. 9.18. – *Fonctions calculées avec une PROM.*

On note que la mémoire PROM contient la forme canonique c'est-à-dire le tableau de vérité d'une fonction. Une réalisation de fonction par une PROM n'implique donc aucun calcul booléen.

3.4.4.2. Réalisation des fonctions E, M, C, R avec une PROM

La plus petite PROM à fusibles est une mémoire de 32 mots de 8 eb. On applique e, m, c, p sur les entrées adresse et dans chaque mot on inscrit les valeurs des fonctions E, M, C, R figurant dans le tableau de vérité de la figure 9.5. On donne à la figure 9.19 le câblage de la PROM et son contenu. On voit que celle-ci est largement sous utilisée.

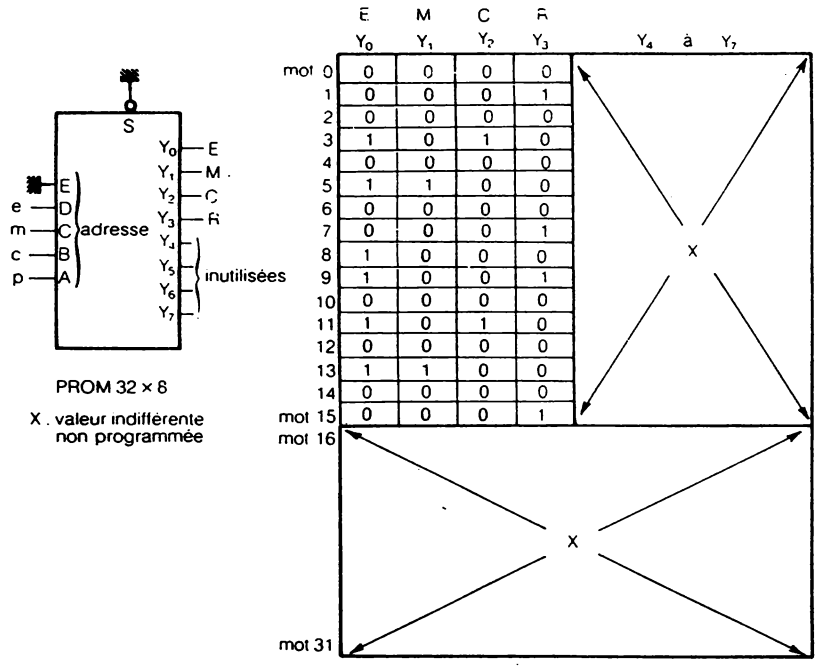


Fig. 9.19. — *Câblage et contenu de la PROM pour les fonctions E, M, C, R.*

4. ÉTUDE D'UN SYSTÈME COMPLEXE

4.1. Spécification

On dispose d'une informations binaire à 16 eb x_0, x_1 ... x_{15} et l'on souhaite connaître avec un temps de réponse très court le nombre d'eb en 1 dans cette information. Plusieurs procédés permettent d'atteindre ce résultat :

— un procédé séquentiel, qui utilise des registres, des compteurs pilotés par des horloges. Comme nous n'avons pas encore étudié ces types de circuits et leur fonctionnement nous ne détaillerons pas plus cette solution. La figure 9.20 en donne seulement le principe. On teste successivement chacun des eb de la configuration et l'on ajoute 1 à un compte d'eb,

si l'eb testé est à 1. Quand tous les eb sont passés en revue, le compte contient la valeur recherchée. Cette solution nécessite 16 examens successifs ce qui est trop long pour le problème posé.

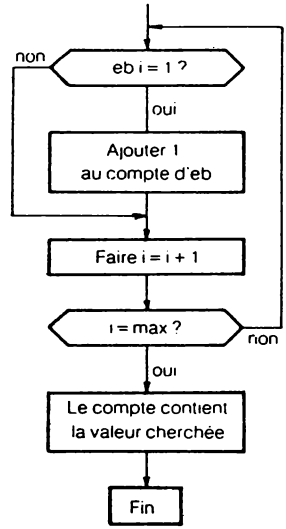


Fig. 9.20. — *Comptage de 1 : méthode séquentielle.*

— un procédé combinatoire dans lequel on considère les 5 valeurs A, B, C, D, E, donnant le compte d'eb comme des fonctions combinatoires des 16 variables d'entrée. Il existe ainsi 65536 combinaisons d'entrée. L'approche directe, avec tableau de vérité, comme dans le cas de systèmes simples, ne peut être envisagée.

4.2. Analyse

On cherche à faire apparaître comme indiqué à la figure 9.1 des fonctions combinatoires connues, ou des sous-systèmes calculables par tableau de vérité.

Une idée est de fractionner les 16 eb en groupes de petite taille, de calculer de manière combinatoire pour ces groupes le nombre d'eb en 1 et d'additionner les résultats partiels par des additionneurs.

La figure 9.21 montre un découpage pour des groupes de 3 eb et qui tire partie de l'entrée de retgenue des additionneurs. Si l'on place un des 16 eb sur cette entrée on ajoute 1 au résultat partiel si l'eb vaut 1 et 0 dans le cas contraire.

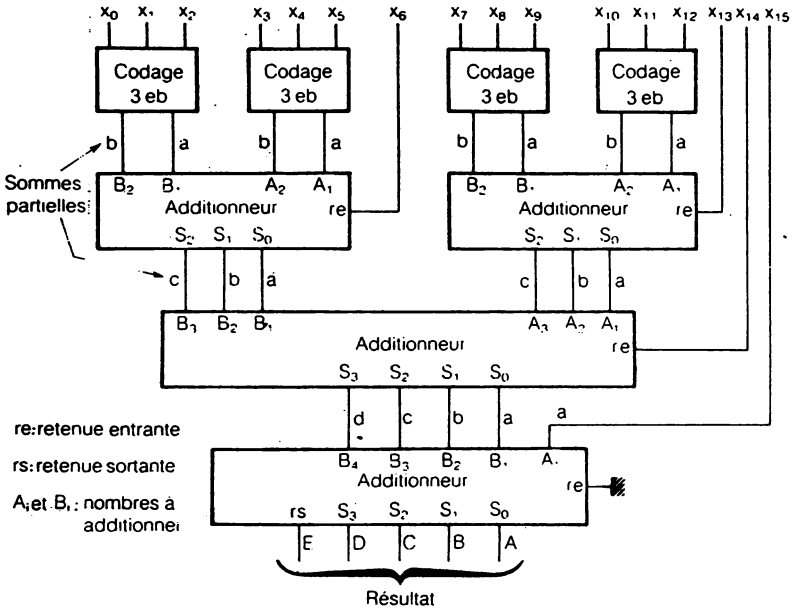


Fig. 9.21. — *Calcul du nombre d'eb en 1 groupe de 3 : Principe.*

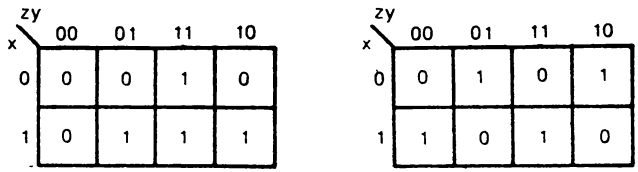
Examinons comment réaliser le codage des groupes d'eb.

4.3. Codage des groupes d'eb

4.3.1. Codage par tableau de vérité

Pour trois eb la fonction de codage est simple (fig. 9.22).

Variables			Somme	
x	y	z	a	b
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$b = \Sigma(3, 6, 7, 5)$
$b = zy + yx + xz$

$a = \Sigma(1, 2, 7, 4)$
$a = z \oplus y \oplus x$

Fig. 9.22. — *Table de vérité et calcul de a et b.*

Ces fonctions peuvent se calculer par des portes SSI, des portes de circuits programmables ou un multiplexeur à deux entrées d'adresse.

4.3.2. Codage avec additionneur

Une autre solution, utilisant uniquement des additionneurs consiste à utiliser les deux premiers étages d'un additionneur 4 eb et la retenue entrante (fig. 9.23).

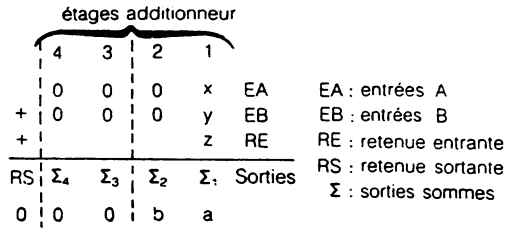


Fig. 9.23. — *Codage de 3 eb par un additionneur.*

On câble x y sur le premier étage, 00 sur le second et z sur la retenue. Les sorties Σ_1 et Σ_2 donnent le codage. La seconde moitié de l'additionneur

est disponible et n'est pas influencée par les deux premiers étages, car il ne peut y avoir de retenue générée par l'étage de rang 2 (deux zéros appliquées).

Ce même circuit peut servir à faire un codage de 4 eb en tirant parti de la seconde moitié inutilisée. On additionne alors le résultat partiel (*a b*) à un quatrième eb (*t*). Le codage à 4 eb apparaît alors sur $\Sigma_3\,\Sigma_4$ et la retenue sortante (fig. 9.24). La figure 9.24 donne le schéma d'ensemble ainsi obtenu.

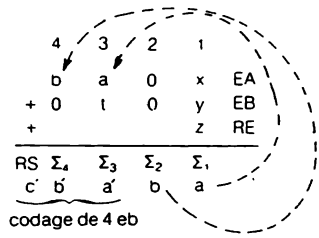


Fig. 9.24. — *Codage de quatre variables xyzt.*

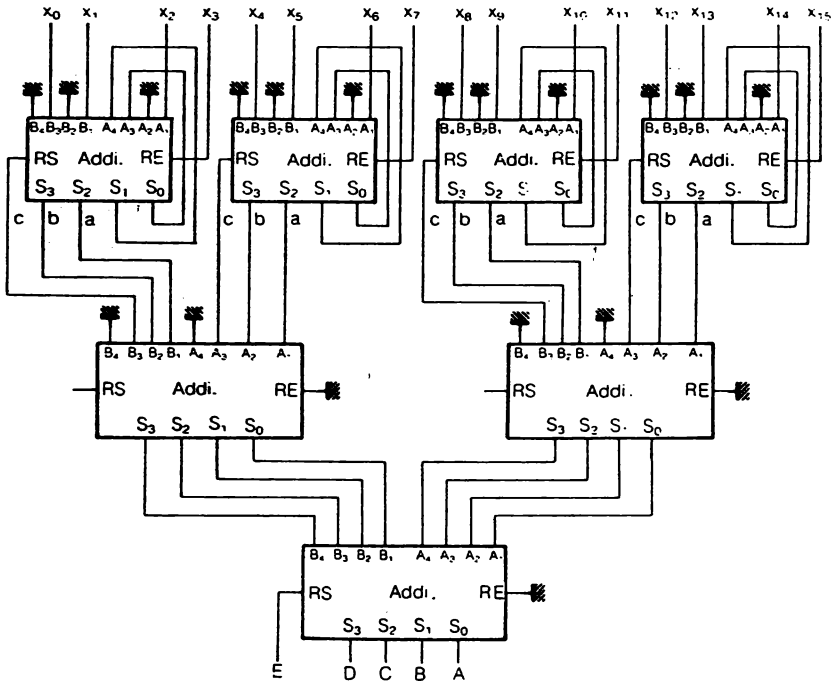


Fig. 9.25. — *Calcul du nombre d'eb en 1 dans un groupe de 16 eb : solution avec additionneurs.*

4.3.3. Solution avec mémoires PROM et additionneurs

Enfin une dernière solution, plus élégante et compacte que les précédentes consiste à coder les eb par une mémoire PROM. Il existe des PROM à 256 mots de 4 eb. Il suffit donc d'en utiliser deux et d'additionner les deux résultats partiels (fig. 9.26).

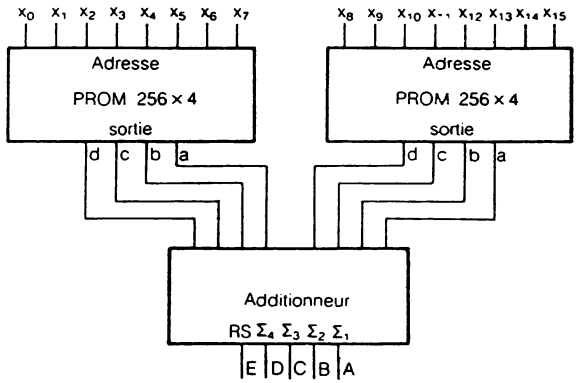


Fig. 9.26. — *Codage par des PROM.*

4.4. Remarque sur les solutions retenues

Dans le cas de systèmes complexes la diversité des solutions possibles est grande. Il n'existe pas de solution type et c'est au concepteur d'effectuer une analyse détaillée du problème posé en ayant toujours en arrière plan les possibilités offertes par les circuits MSI ou LSI existants.

5. COMPARAISON ENTRE LES DIFFÉRENTES MÉTHODES DE RÉALISATION

Le tableau de la figure 9.27 compare les différentes façons de réaliser les fonctions logiques. Les comparaisons portent sur :

- la nécessité ou non de faire des calculs booléens,
- le coût des composants utilisés,
- le nombre de boîtiers nécessaires,

— la vitesse de traitement (temps de calcul des fonctions),
— la consommation,
— la souplesse de modification,
— la nécessité d'un matériel annexe de programmation,
— le domaine d'emploi privilégié.

Bien entendu, les paramètres donnés dans ce tableau ne sont pas immuables. Les prix des PAL en particulier sont en baisse depuis qu'il existe plusieurs sources. Leur vitesse s'améliore aussi. Le coût très bas des portes ne doit pas masquer qu'il faut en mettre beaucoup pour obtenir l'équivalent d'une fonction réalisée par un circuit MSI et LSI et qu'interviennent alors d'autres paramètres comme la surface sur la carte supportant les boîtiers, la difficulté de tracé du circuit imprimé, etc...

La vitesse de traitement élevée au niveau des portes est pratiquement annulée par le fait que plusieurs couches de portes sont nécessaires pour réaliser des fonctions équivalentes aux circuits MSI ou PAL.

En ce qui concerne la consommation elle est en faveur des portes et MSI par rapport aux PAL et PROM pour les exemples que nous avons présentés.

La souplesse de modification est très grande pour les PAL et PROM. Il ne faut cependant pas avoir beaucoup d'illusions sur les possibilités de modification d'un produit réalisé à plusieurs milliers ou dizaine de milliers d'exemplaires. S'il est réalisé à partir de portes ou de MSI il faut intervenir sur le câblage et donc sur les circuits imprimés (coupures de pistes, fils ajoutés, éventuellement reprise du dessin du circuit imprimé et réalisation de nouvelles cartes) et le coût d'une modification devient rapidement très élevé.

Si la partie à modifier est réalisée à partir de PAL, PROM ou circuit à la demande on ne touche pas au circuit imprimé (sauf si de nouvelles fonctions ou variables sont introduites) par contre il faut programmer de nouveaux boîtiers pour les substituer aux anciens. Finalement les produits rentrent généralement en usine et les coûts de main d'œuvre, transport, etc... dépassent largement ceux des composants à changer.

Enfin il semble d'après ce tableau que les PAL et les PROM ont le même domaine d'emploi. En fait, il existe rarement des fonctions complexes (beaucoup de monômes) de nombreuses variables mais plutôt des fonctions relativement simples (peu de monômes) de nombreuses variables. Avec une solution avec PROM, il faut une capacité de 2^n mots s'il y a n variables, la PROM engendrant tous les monômes possibles des n variables. Seuls les

Composant utilisé	Calculs booléens	Coût des boîtiers	Nombre de boîtiers	Vitesse de traitement	Consommation d'énergie par boîtier	Souplesse de modification	Matériel de programmation	Domaine privilégié d'emploi
Portes	oui	très faible 1,5 F / boîtier	élevé	élevée 15 ns/ porte	faible 15 mW	faible	non	peu de fonction fonctions simples peu de variables
PAL	oui	encore élevé 60 F / boîtier	faible	faible 40 ns	élevée 450 à 1 000 mW	très grande	oui	fonctions nombreuses fonctions simples ou complexes peu ou beaucoup de variables
Multiplexeur	non	faible	moyen	faible 43 ns	faible 50 mW	grande	non	peu de fonctions fonctions complexes quelques variables
Décodeur	non	faible	moyen	faible 41 ns	faible 50 mW	grande	non	quelques fonctions simultanées des mêmes variables fonctions complexes quelques variables
PROM	non	moyen	faible	faible 70 ns pour 6349-1	élevée 850 mW	très grande	oui	fonctions nombreuses fonctions simples ou complexes peu ou beaucoup de variables
circuit à la demande	oui	fonction de la quantité	faible	fonction de la technologie	fonction de la complexité	aucune	oui programmation par constructeur	Même domaine que PAL et PROM leur intérêt réside uniquement dans un calcul économique

Fig. 9.27. — *Comparaison des outils de réalisation des fonctions booléennes.*

monômes utilisés donnent lieu à l'inscription d'un 1. Il y a donc un certain gâchis. Par contre avec des PAL seuls les monômes utiles sont calculés. Ce qui permet, avec la même complexité interne qu'une PROM de calculer des fonctions booléennes d'un plus grand nombre de variables.

Chapitre 10

LES FONCTIONS SÉQUENTIELLES SIMPLES

1. COMPORTEMENT DES SYSTÈMES COMBINATOIRES ET SÉQUENTIELS

Soit un système combinatoire ayant plusieurs entrées et plusieurs sorties. Supposons qu'initialement les variables présentées sur les entrées aient des valeurs stables. Les fonctions de sorties ont également des valeurs stables, dépendant du tableau de vérité qui lie les entrées et les sorties. A un instant donné, modifions l'état des entrées.

Les sorties prennent d'abord des états transitoires, de courte durée (quelques retards de portes) qui dépendent:

- Des retards propres à chaque composant constituant le logigramme des fonctions (temps t_{PLH} et t_{PHL}).
- De la constitution même du logigramme: nombre de portes ou circuits connectés en cascade.
- Des petits écarts de phase entre les variables d'entrée au moment du changement d'état des entrées. En effet, même pour un changement "simultané", il y a des petites différences de phase car les signaux proviennent des circuits possédant des retards légèrement différents.

Compte tenu de l'incertitude associée aux retards des circuits (on ne connaît que des valeurs extrêmes des retards, minimum ou maximum, pour une famille de circuits et jamais pour un circuit particulier, à moins de l'avoir mesuré ou conçu spécialement), on ne peut pas prévoir le comportement précis des sorties dans l'état transitoire. On peut seulement apprécier sa durée maximum. Par exemple, dans un additionneur, on n'utilise le résultat qu'après avoir laissé le temps à la retenue de se propager dans le pire cas. Lorsque la phase transitoire est terminée, les sorties prennent des valeurs stables, déterminées à partir des entrées stabilisées par le tableau de vérité. On peut dire, en première approximation, (en négligeant la phase transitoire) que les sorties d'un système combinatoire ne dépendent pas du temps, mais seulement des états appliqués aux entrées.

Dans les circuits séquentiels, que nous allons étudier maintenant, le fonctionnement dépend essentiellement du temps. Si l'on modifie les entrées d'un système séquentiel, les sorties passent également par un état transitoire, dû au retard des circuits et au logigramme, avant d'atteindre un état stable. Mais la différence importante avec les circuits combinatoires, c'est que l'état stable *dépend de l'ordre* dans lequel les signaux d'entrée sont appliqués. En particulier, les petits écarts de phase entre les signaux d'entrée lors d'une variation "simultanée" ne sont pas négligeables, car ils traduisent un échelonnement particulier des signaux d'entrée. Pour obtenir une combinaison de sortie déterminée dans un système séquentiel, il faut veiller à présenter les signaux d'entrée dans un ordre déterminé en respectant des contraintes de temps (durée des signaux, écarts entre signaux, fréquence maximale, etc...).

2. LES FONCTIONS SÉQUENTIELLES SIMPLES

L'étude et la synthèse optimale des fonctions combinatoires ont donné lieu à de nombreux travaux théoriques. Des efforts comparables ont été portés sur les fonctions séquentielles. Les résultats de ces recherches peuvent intéresser les théoriciens de la logique mais pas à notre avis, ceux qui conçoivent des systèmes. Aussi pensons-nous qu'une approche simple, sans mathématique ni théorie complexe, étudiant les circuits d'une manière descriptive (comment réagit un circuit à des actions sur les entrées, que mettre sur les entrées pour obtenir des sorties déterminées, quelles sont les contraintes d'utilisation d'un circuit, etc...), suffit largement. Dans la suite, nous nous occuperons surtout des caractéristiques *externes* des circuits, sans entrer dans le fonctionnement *interne*, sauf s'il est particulièrement

simple. De même, sauf exception, nous ne donnerons que les schémas symboliques des circuits ("boîte noire" avec entrées et sorties). L'intérieur des circuits est l'affaire des concepteurs de circuits à qui cet ouvrage n'est pas destiné. Notre but consiste plutôt à montrer comment utiliser des *circuits existants* pour concevoir des systèmes.

Nous étudierons d'abord dans ce chapitre quatre fonctions séquentielles de base :

– *Le bistable.*

Ce circuit a deux états stables de sortie, 1 ou 0 commandables à partir des entrées.

– *La bascule.*

Ce circuit a deux états stables de sortie 1 ou 0 comme le bistable, commandables à partir des entrées. Mais la constitution plus complexe des bascules permet des montages avec rétroactions et automodifications interdits avec les bistables.

– *Le monostable.*

Ce circuit possède une position de repos. Quand il reçoit des signaux appropriés aux entrées, il quitte sa position de repos pour donner une impulsion de durée déterminée en sortie et revenir ensuite à sa position de repos.

– *L'astable ou oscillateur.*

Ce circuit n'a pas de position de repos. Il passe indéfiniment d'un état 1 à un état 0, avec une période T sans commande extérieure.

3. LE BISTABLE RS

3.1. Schéma

La figure 10.1 donne le schéma de principe du bistable RS. Ce circuit comprend deux entrées R et S, servant respectivement à mettre le bistable dans l'état 0 (R : Reset ou remise à zéro) ou dans l'état 1 (S : Set ou mise à 1). Il a deux sorties Q et $\overline{Q}$ inverses l'une de l'autre.

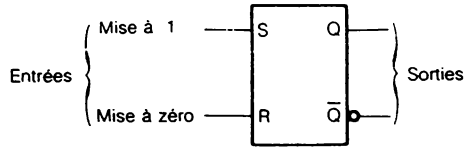


Fig. 10.1. — *Schéma de principe du bistable RS.*

3.2. Fonctionnement statique

Le fonctionnement de ce circuit se décrit par une table de vérité (fig. 10.2) qui présente la particularité d'avoir en entrée non seulement les variables d'entrée R et S (comme pour une fonction combinatoire) mais aussi la fonction de sortie Q. A partir de R, S et Q, la table donne la nouvelle valeur de la sortie, notée Q^+. Cette table s'interprète ainsi :

- Si $R = S = 0$, le bistable conserve son état ancien.
- Si $R = 1$, $S = 0$, le bistable passe à 0, quel que soit son état ancien.
- Si $R = 0$, $S = 1$, le bistable passe à 1, quel que soit son état ancien.
- Le cas $R = S = 1$ ne s'exploite pas, car cela revient à mettre le bistable à la fois dans l'état 1 et dans l'état 0.

Cas	R	S	Q	Q^+	Remarques
0	0	0	0	0	Maintien
1	0	0	1	1	Maintien
2	0	1	0	1	Mise à 1
3	0	1	1	1	Mise à 1
4	1	0	0	0	Mise à 0
5	1	0	1	0	Mise à 0
6	1	1	0	X	non exploité
7	1	1	1	X	non exploité

Fig. 10.2. — *Table de vérité d'un bistable RS.*

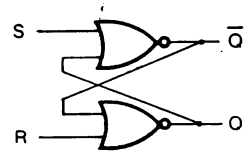


Fig. 10.3. — *Bistable RS réalisé avec des portes NOR.*

3.3. Étude du schéma du bistable RS réalisé avec des portes NOR

Ce bistable ayant un schéma particulièrement simple, on peut facilement vérifier la concordance entre la table de vérité (fig. 10.2) et le schéma (fig. 10.3).

Cas 0 : $R = S = Q = 0$. On a :

$$\bar{Q}^+ = \overline{Q + S} = \overline{0 + 0} = 1$$

$$Q^+ = \overline{\bar{Q} + R} = \overline{1 + 0} = 0$$

Si un bistable RS se trouve dans l'état 0, il conserve cet état si les entrées R et S sont maintenues à 0.

Cas 1 : $R = S = 0$, $Q = 1$

$$\bar{Q}^+ = \overline{Q + S} = \overline{1 + 0} = 0$$

$$Q^+ = \overline{\bar{Q} + R} = \overline{0 + 0} = 1$$

Si un bistable RS se trouve dans l'état 1, il conserve cet état si les entrées R et S sont maintenues à 0.

Cas 2 : $R = 0$, $S = 1$, $Q = 0$

$$\bar{Q}^+ = \overline{Q + S} = \overline{0 + 1} = 0$$

$$Q^+ = \overline{\bar{Q} + R} = \overline{0 + 0} = 1$$

Si un bistable RS est dans l'état 0 et si on lui applique $S = 1$, il passe dans l'état 1.

Cas 3 : $R = 0$, $S = 1$, $Q = 1$

$$\bar{Q}^+ = \overline{Q + S} = \overline{1 + 1} = 0$$

$$Q^+ = \overline{\bar{Q} + R} = \overline{0 + 0} = 1$$

Si un bistable RS est dans l'état 1 et si on lui applique $S = 1$, il reste dans l'état 1.

Cas 4 : $R = 1$, $S = 0$, $Q = 0$

$$Q^+ = \overline{\bar{Q} + R} = \overline{1 + 1} = 0$$

$$\bar{Q}^+ = \overline{Q + S} = \overline{0 + 0} = 1$$

Si un bistable RS est dans l'état 0 et si on lui applique $R = 1$, il conserve son état.

Cas 5 : $R = 1$, $S = 0$, $Q = 1$

$$Q^+ = \overline{\bar{Q} + R} = \overline{0 + 1} = 0$$

$$\bar{Q}^+ = \overline{\bar{Q} + S} = \overline{0 + 0} = 1$$

Si un bistable *RS* est dans l'état 0 et si on lui applique $R = 1$ il passe à l'état 0.

Cas 6 et 7 : $R = S = 1$, $Q = 0$ ou 1

Il s'agit de cas anormaux d'exploitation où l'on souhaite que le bistable prenne simultanément l'état 0 et l'état 1. Ce n'est pas possible. Dans ce cas, les deux sorties Q et $\overline{Q}$ ne sont plus inverses. On a :

$$Q^+ = \overline{\overline{Q} + R} = \overline{X + 1} = 0$$

$$\overline{Q}^+ = \overline{Q + S} = \overline{X + 1} = 0$$

Quand on applique simultanément R et S à 1 pour un bistable *RS* conçu avec des portes NOR, les deux sorties sont égales $Q = \overline{Q} = 0$. Ce cas d'exploitation est déconseillé car les deux sorties ne sont plus complémentaires l'une de l'autre, ce qui conduit à de fausses interprétations du fonctionnement.

3.4. Fonctionnement dynamique du bistable réalisé avec des portes NOR

Jusque là, nous avons appliqué des états sur R et S. Dans la pratique, on utilise des impulsions de courte durée, sans recouvrement entre elles. Envoyons successivement une impulsion sur l'entrée S, puis une autre sur l'entrée R, d'un bistable initialement à l'état 0.

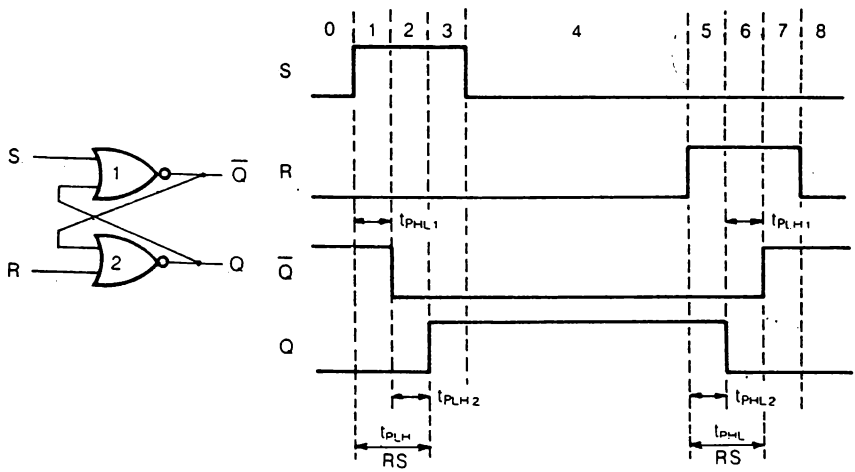


Fig. 10.4. — *Fonctionnement dynamique d'un bistable RS.*

La figure 10.4 montre le diagramme des temps qui en résulte.

- *Situation 0*: $S = R = 0$, le bistable conserve son état $Q = 0$, $\overline{Q} = 1$.
- *Situation 1*: $S = 1$, $R = 0$. La sortie $\overline{Q}$ réagit à l'entrée S. Elle passe du niveau 1 au niveau 0 au bout du temps de retard t_{PHL} de la porte 1. La sortie Q n'a pas encore réagi.
- *Situation 2*: $S = 1$, $R = 0$. La porte 2 reçoit $\overline{Q} = 0$ et $R = 0$, sa sortie passe à 1 au bout d'un temps t_{PLH} de la porte 2.
- *Situation 3*: $S = 1$, $R = 0$. La porte 1 reçoit $S = 1$ et $Q = 1$, ce qui confirme l'état 0 en sortie $\overline{Q}$. A partir de cet instant, la commande $S = 1$ peut disparaître, la sortie $Q = 1$ ayant pris le "relais".
- *Situation 4*: $R = S = 0$. Le bistable conserve son état $Q = 1$, $\overline{Q} = 0$.
- *Situation 5*: $R = 1$, $S = 0$. La sortie Q réagit à l'entrée R. Elle passe à 0 au bout du temps t_{PHL} de la porte 2. La sortie $\overline{Q}$ n'a pas encore réagi.
- *Situation 6*: $R = 1$, $S = 0$. La porte 1 reçoit $Q = 0$ et $S = 0$. La sortie $\overline{Q}$ passe à 1 au bout du temps t_{PLH} de la porte 1.
- *Situation 7*: $R = 1$, $S = 0$. La porte 2 reçoit $R = 1$, $\overline{Q} = 1$ ce qui confirme l'état 0 en sortie Q. A partir de cet instant, la commande R peut disparaître, $\overline{Q}$ ayant pris le "relais".
- *Situation 8*: On se retrouve dans la situation 0.

3.5. Remarques sur le fonctionnement dynamique

De l'analyse précédente on peut déduire quelques propriétés du régime dynamique:

- La durée minimale d'une impulsion R ou S est de $t_{PHL} + t_{PLH}$ soit 30 *ns* avec des portes NOR 74 LS 02.
- Pendant les instants de commutation (situations 2 et 6), les deux sorties Q et $\overline{Q}$ ne sont pas inverses l'une de l'autre mais égales: $Q = \overline{Q} = 0$. Si l'on établit des fonctions à partir de Q et $\overline{Q}$, il faudra veiller à ce qu'elles ne soient pas actives pendant cette période de temps, car alors deux fonctions différentes, dépendant de Q pour l'une et de $\overline{Q}$ pour l'autre, pourraient agir au même moment quand $Q = \overline{Q}$.
- Le temps de basculement du bistable ($t_{PHL\ RS}$ ou $t_{PLH\ RS}$ de la figure 10.4) est égal à la somme des temps t_{PHL} et t_{PLH} des portes NOR utilisées soit 30 *ns* maximum pour des portes 74 LS 02.

– Si l'on applique des impulsions R et S simultanément en 1 (cas anormal d'exploitation), les deux sorties Q et $\bar{Q}$ prennent la même valeur $Q = \bar{Q} = 0$ pendant tout le temps où $R = S = 1$. Quand on ramène R et S à 0, trois cas sont à envisager :

- R passe à 0 légèrement avant S

  $RS = 11 \rightarrow RS = 01 \rightarrow RS = 00 \rightarrow Q = 1$

  C'est S qui impose l'état 1 du bistable.

- *S passe à 0 légèrement avant R*

  $RS = 11 \rightarrow RS = 10 \rightarrow RS = 00 \rightarrow Q = 0$

  C'est R qui impose l'état 0 du bistable.

- R et S passent à 0 en même temps

  $RS = 11 \rightarrow Q = \bar{Q} = 0$

  $RS = 00 \rightarrow Q = \overline{R + \bar{Q}} = \overline{0 + 0} = 1, \bar{Q} = \overline{S + Q} = \overline{0 + 0} = 1$

  Les deux sorties Q et $\bar{Q}$ passent à 1. Au bout d'un retard de portes on a :

  $RS = 00, Q = \bar{Q} = 1$

  $Q = \overline{R + \bar{Q}} = \overline{0 + 1} = 0, \bar{Q} = \overline{S + Q} = \overline{0 + 1} = 0$

  Les deux sorties reviennent à 0. Le même cycle recommence. On a donc une suite indéfinie d'oscillations, on dit que le bistable se trouve dans l'état métastable. En pratique il finit par quitter cet état car il existe toujours une dissymétrie qui impose un état stable 0 ou 1 sur l'une des sorties. On ne peut pas définir la durée de cet état métastable car il dépend des caractéristiques internes du bistable. En conclusion il ne faut pas exciter un bistable par des impulsions simultanées.

3.6. Bistable RS réalisé avec des portes NAND

La figure 10.5 montre le schéma symbolique et de détail d'un bistable RS réalisé avec des NAND. En fait, comme les entrées agissent avec des 0, il s'agit plutôt d'un bistable $\bar{R}\bar{S}$.

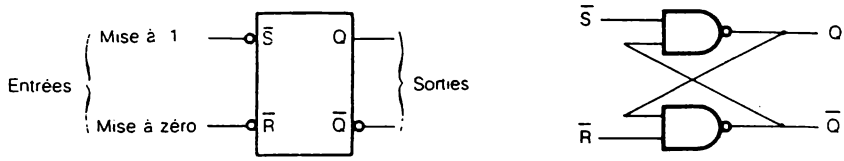


Fig. 10.5. — *Schémas de bistables $\bar{R}\bar{S}$ réalisé avec des NAND.*

Son fonctionnement statique se déduit du tableau de vérité de la figure 10.2 en inversant le contenu des deux colonnes R et S.

$\bar{R} = \bar{S} = 1$, conservation de l'état ancien.

$\bar{R} = 0, \bar{S} = 1$, le bistable passe à 0

$\bar{R} = 1, \bar{S} = 0$, le bistable passe à 1

$\bar{R} = \bar{S} = 0$, cas d'exploitation interdit

De l'étude du fonctionnement dynamique du bistables $\bar{R}\bar{S}$, on tire les mêmes conclusions: durée minimum d'impulsion égale à $t_{PLH} + t_{PHL}$, non complémentarité des sorties pendant la période transitoire de commutation mais avec $Q = \bar{Q} = 1$ au lieu de 0 pour un bistable réalisé avec des portes NOR, temps de basculement et état métastable.

4. LE BISTABLE D

Ce bistable comprend deux entrées D (comme Data ou donnée) et CK (clock ou horloge) et deux sorties inverses Q et $\bar{Q}$. La figure 10.6 donne les schémas symbolique et de détail du bistable D et les figures 10.7 et 10.8 les tableaux de vérité et le diagramme des temps de fonctionnement.

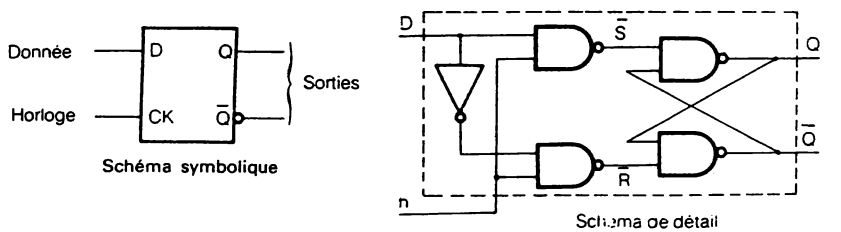


Fig. 10.6. — *Schémas du bistable D.*

D	CK	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tableau complet

D	CK	Q^+	Remarques
X	0	Q	Maintien
D	1	D	Affichage

Tableau réduit

Fig. 10.7. — *Tableaux de vérité du bistable D.*

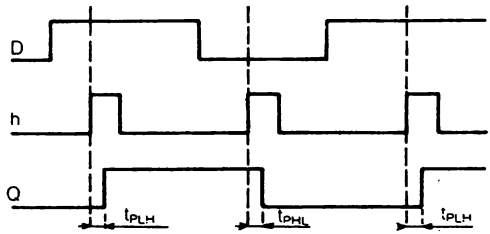


Fig. 10.8. — *Diagramme des temps du bistable D.*

Le tableau de vérité réduit s'interprète ainsi :

— Si $CK = 0$, le bistable conserve son état ancien, noté Q, quelle que soit l'information affichée sur D.

— Si $CK = 1$, le bistable prend l'état affiché sur D.

Le signal d'horloge peut être périodique ou non. Il joue le rôle d'un signal de validation pour la donnée entrant dans le bistable.

5. UTILISATION DES BISTABLES

On n'utilise pratiquement plus les bistables *RS* et les bistables *D* comme composants d'un montage. Il n'existe d'ailleurs quasiment pas de circuits intégrés de ce type. On rencontre parfois les bistables *RS* dans des schémas de composants intégrés complexes. Quand aux bistables *D*, ils servent surtout à constituer des registres (voir chapitre sur les registres de mémorisation).

6. RETROACTIONS DANS UN SYSTÈME SÉQUENTIEL

6.1. Limite d'emploi des bistables

Soit un système séquentiel (fig. 10.9) comprenant :

- Deux bistables D, Q_A et Q_B.
- Un réseau combinatoire calculant les fonctions D_A et D_B à partir de Q_A, Q_B et de trois entrées E_1, E_2, E_3. Ces fonctions D_A et D_B permettent d'obtenir l'état futur de Q_A et Q_B. Les deux bistables sont commandés par le même signal d'horloge h.
- Un réseau combinatoire calculant les fonctions f_1, f_2, f_3, f_4 de sortie dépendant de Q_A, Q_B, E_1, E_2, E_3.

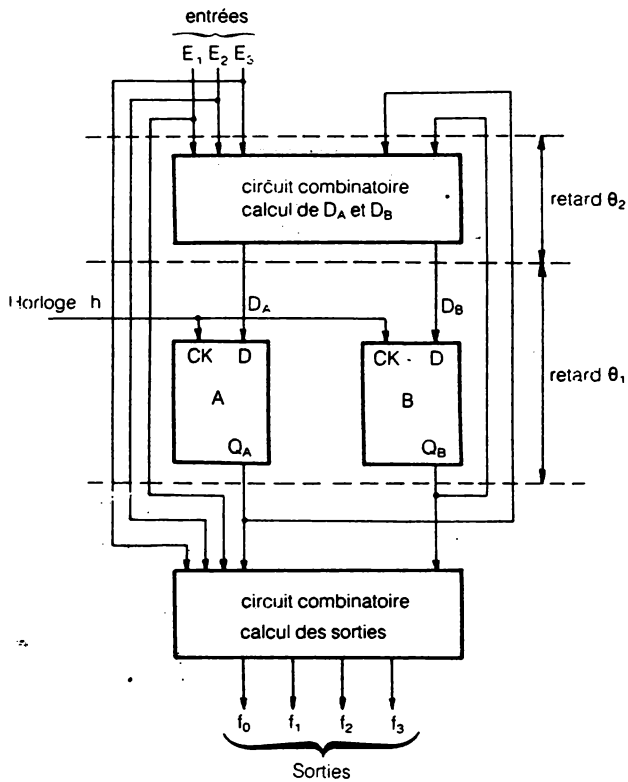


Fig. 10.9. — *Exemple de système séquentiel avec rétroaction.*

On souhaite qu'à chaque signal d'horloge h, le système évolue en fonction des états des entrées et de l'état présent de Q_A et Q_B pour donner un nouvel état Q_A^+ et Q_B^+ et des fonctions de sorties f_1, f_2, f_3 et f_4. Un tel système se décrit par un tableau de vérité comme celui de la figure 10.10 (son contenu est quelconque).

Entrees			Etat present		Etat futur		Excitation		Fonctions de sortie			
E_1	E_2	E_3	Q_A	Q_B	Q_A^+	Q_B^+	D_A	D_B	f_0	f_1	f_2	f_3
0	0	0	0	0	0	1	0	1	1	0	0	1
1	1	1	1	1	1	0	1	0	0	0	1	1

Fig. 10.10. — *Exemple de tableau de vérité d'un système.*

Dans ce tableau, il est sous entendu que les évolutions n'ont lieu qu'au signal horloge h. Or comme le montre le diagramme des temps de la figure 10.11, le système ne fonctionne pas correctement.

A l'origine, avant l'impulsion d'horloge (niveau bas), les bistables A et B se trouvent dans un état Q_{A1} et Q_{B1}. Les fonctions D_{A1} et D_{B1} sont présentes mais n'agissent pas puisque l'horloge h vaut 0.

L'horloge h apparaît. Les fonctions D_{A1} et D_{B1} agissent sur Q_A et Q_B pour donner un état 2, Q_{A2} et Q_{B2} conformément au tableau de vérité, au bout d'un temps θ_1, temps de commutation des bistables.

Le nouvel état Q_{A2}, Q_{B2} traverse le circuit combinatoire calculant D_A et D_B de sorte qu'après un temps θ_2 de nouvelles fonctions, D_{A2} et D_{B2} sont disponibles. Comme l'horloge h vaut toujours 1 (sa durée dépasse largement le retard des circuits), ces fonctions D_{A2} et D_{B2} provoquent l'affichage d'un nouvel état Q_{A3} et Q_{B3} un temps θ_1 plus tard. Et ainsi de suite tant que dure l'horloge. Le dernier état sera celui affiché juste avant la fin de l'horloge (Q_{A4} et Q_{B4} pour la figure 10.11).

Le système ne fonctionne pas car au lieu d'une transition unique, on a des transitions en cascade. De plus, les retards θ_1, θ_2 ne sont pas exactement les mêmes pour Q_A, Q_B, D_A, D_B de sorte que les fonctions D_A, D_B, Q_A, Q_B sont décalées de manière non prévisibles dans le temps. Il en résulte des états non contrôlables pour Q_A et Q_B. Le même problème se poserait si l'on remplaçait les bistables D par des bistables RS.

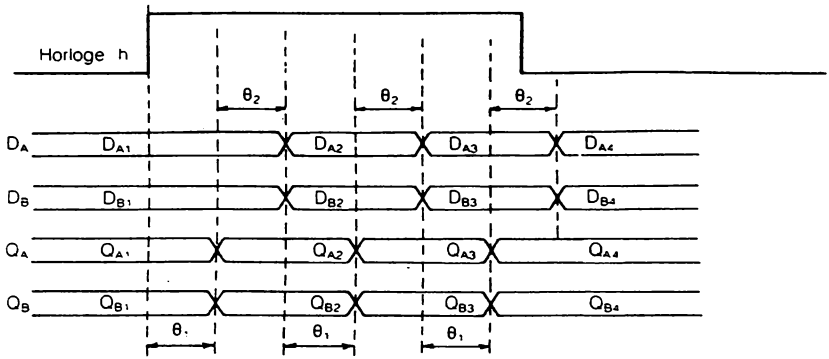
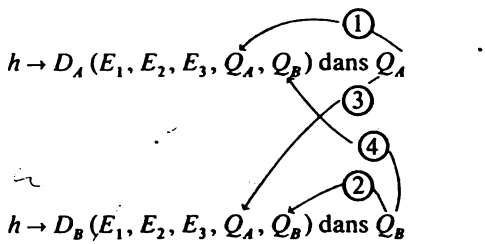


Fig. 10.11. — *Diagramme d'un système séquentiel avec rétroaction.*

6.2. Les rétroactions nécessitent des circuits complexes

Les équations de commande du système précédent sont de la forme:



Ces équations s'interprètent de la manière suivante: quand le signal h est présent, la fonction D_A des variables E_1, E_2, E_3, Q_A, Q_B est transféré dans Q_A; quand h vaut 0, le transfert n'a pas lieu et Q_A garde sa valeur. Il en est de même pour Q_B.

Les flèches désignées ① et ② indiquent une *automodification*: la valeur de Q_A par exemple, une fois modifiée est prise en compte dans la fonction D_A pour agir sur Q_A, donnant un nouvel état, lui aussi pris en compte, et ainsi de suite.

Les flèches désignées ③ et ④ indiquent une *rétroaction croisée*: la valeur de Q_A modifiée intervient sur l'affichage de Q_B, tandis que Q_B modifié intervient sur l'affichage de Q_A, qui agit à nouveau sur l'affichage de Q_B et ainsi de suite.

La cause du mauvais fonctionnement réside dans le fait qu'un état de sortie peut réagir immédiatement dans le réseau d'entrée pour donner un nouvel état de sortie. Pour éviter cela, il faut *isoler* la sortie du bistable de son entrée, dès que la sortie a commuté et ne la prendre en compte pour provoquer un nouvel état bistable qu'au prochain signal d'horloge. Plusieurs solutions ont été utilisées :

1) Le bistable est remplacé par deux bistables en cascade, appelés maître et esclave, commandés par deux horloges différentes, l'une servant au bistable de sortie (esclave) l'autre au bistable d'entrée (maître). Quand le bistable de sortie a basculé sous l'influence de h_2, son état n'influence pas immédiatement le bistable d'entrée (fig. 10.12) : il faut attendre la prochaine impulsion h_1.

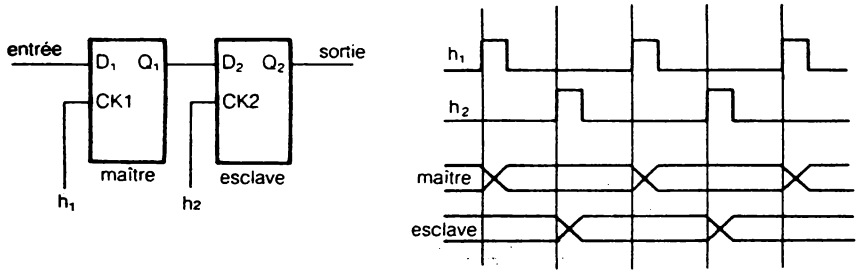


Fig. 10.12. — *Circuit maître esclave à deux temps.*

2) Le bistable est remplacé par deux bistables maître et esclave, pilotés par une même horloge. Les seuils de tension de commande sur les entrées CK_1 et CK_2 des deux bistables sont différentes de sorte que l'horloge unique est vue comme deux horloges distinctes.

3) Le bistable est remplacé par un montage à trois bistables ou par un montage séquentiel complexe, piloté par une seule horloge. Les données d'entrée ne sont enregistrées que pendant une période très brève autour d'un front montant ou descendant de l'horloge. Le circuit reste ensuite insensible aux données d'entrée s'il n'y a pas de transition d'horloge.

A l'heure actuelle, la solution 3 est la plus utilisée, la solution 2 se rencontre dans les circuits anciens et la solution 1 n'est plus utilisée. Ces circuits plus complexes que les bistables qui permettent les automodifications et les rétro-actions portent le nom de *bascules*.

Seuls trois types de bascules ont été développés :

— La bascule *RS* qui n'est pas employée seule mais à l'intérieur de composants séquentiels plus complexes.

– La bascule *JK*.

– La bascule *D*.

Nous examinerons dans ce qui suit uniquement les bascules *JK* et *D*.

7. LA BASCULE JK

7.1. Schémas symboliques

La figure 10.13 donne deux schémas symboliques de bascules *JK*. Nous ne donnons pas les schémas de détail car ils sont trop complexes et parce que l'étude du fonctionnement interne n'apporterait rien à l'utilisateur d'une telle bascule. La table de vérité et les paramètres électriques et temporels définissent suffisamment le comportement extérieur de la bascule.

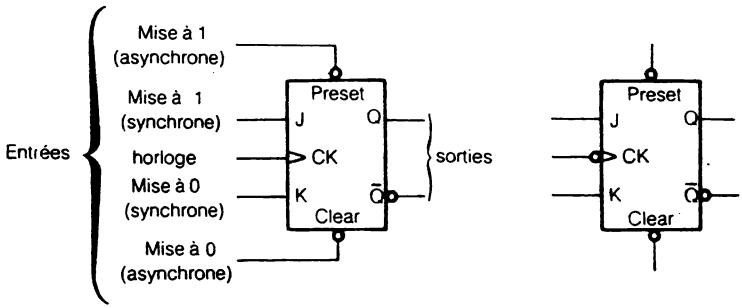


Fig. 10.13 – *Schémas symboliques de bascule JK.*

Le rond sur les entrées Preset et Clear indique qu'elles agissent avec un 0. Le rond en sortie $\overline{Q}$ indique que l'on a un 0 lorsque la bascule est en 1. Le triangle sur l'entrée *CK* indique que cette entrée agit sur la sortie au voisinage d'un front d'horloge. Si un rond est placé avant le triangle, le signal appliqué sur l'entrée agit au voisinage d'un front descendant d'horloge et s'il n'y a pas de rond, il agit au voisinage du front montant.

La bascule *JK* comporte :

– Une entrée Preset de mise à 1, asynchrone, c'est-à-dire indépendante de l'horloge, active en inverse.

– Une entrée Clear de mise à 0, asynchrone, active en inverse.

- Une entrée horloge qui valide les entrées J et K.
- Une entrée J de mise à 1, synchrone, c'est-à-dire n'agissant qu'avec l'horloge, active en direct.
- Une entrée K de mise à 0, synchrone, active en direct.
- Deux sorties Q et $\overline{Q}$ inverses l'une de l'autre.

7.2. Fonctionnement statique

Le tableau de vérité de la figure 10.14 montre le fonctionnement de la bascule JK. Les trois premières lignes correspondent au fonctionnement asynchrone et les cinq suivantes au fonctionnement synchrone. Ce tableau s'interprète ainsi :

- *Cas 0* : Preset = 0 (actif), Clear = 1 (inactif) : la bascule passe à 1 quel que soit l'état de CK, J et K.
- *Cas 1* : Preset : 1 (inactif), Clear = 0 (actif) : la bascule passe à 0, quel que soit l'état de CK, J et K.
- *Cas 2* : Preset : 0 (actif), Clear = 0 (actif). Les deux sorties prennent la valeur 1 ($Q = \overline{Q} = 1$) mais cet état n'est pas stable, quand Preset et Clear retournent à leur état inactif. On a les mêmes phénomènes que pour le bistable RS avec deux commandes simultanées.

	Cas	Preset	Clear	Clock	J	K	Q^+	$\overline{Q}^+$	Remarques
Mode asyn.	0	0	1	X	X	X	1	0	mise à 1
	1	1	0	X	X	X	0	1	mise à 0
	2	0	0	X	X	X	1	1	Etat instable
Mode synch.	3	1	1	↑	0	0	Q	$\overline{Q}$	maintien
	4	1	1	↑	0	1	0	1	mise à 0
	5	1	1	↑	1	0	1	0	mise à 1
	6	1	1	↑	1	1	$\overline{Q}$	Q	Commutation
	7	1	1	0 ou 1	X	X	Q	$\overline{Q}$	maintien

Fig. 10.14. — *Tableau de vérité d'une bascule JK. Action des différentes entrées.*

Ces trois lignes correspondent au fonctionnement d'un bistable RS réalisé avec des portes NAND. Preset joue le rôle de $\overline{S}$ et Clear celui de $\overline{R}$. Pour les lignes suivantes, fonctionnement synchrone, les entrées Preset et Clear sont dans leur état inactif (Preset = Clear = 1).

- *Cas 3* : $J = K = 0$ et front montant d'horloge, la bascule conserve son état.
- *Cas 4* : $J = 0$, $K = 1$ et front montant d'horloge, la bascule passe à 0.
- *Cas 5* : $J = 1$, $K = 0$ et front montant d'horloge, la bascule passe à 1.

Ces trois lignes correspondent au fonctionnement d'un bistable RS réalisé avec des portes NOR, l'action de l'horloge étant mise à part : J joue le rôle de S et K celui de R.

- *Cas 6* : $J = K = 1$ et front montant d'horloge : la bascule commute, c'est-à-dire prend l'état inverse de l'état précédent.
- *Cas 7* : $J = K = X$ et 0 ou 1 sur l'horloge, c'est-à-dire pas de front montant sur l'horloge : la bascule conserve son état.

7.3. Diagramme des temps de fonctionnement

La figure 10.15 montre deux diagrammes des temps : l'un correspondant au mode asynchrone, l'autre au mode synchrone. Ils s'établissent à partir du tableau de vérité.

7.4. Table de vérité d'une bascule JK et table de transition

Du tableau de la figure 10.14, nous pouvons tirer la table de vérité détaillée de la bascule en fonctionnement synchrone, c'est-à-dire la transition Q vers Q^+ obtenue en appliquant une configuration JK pour un état donné Q (fig. 10.16). De cette dernière table, nous déduisons la table de transition de la bascule qui décrit l'action inverse. Connaissant la transition Q vers Q^+ désirée, quelle configuration faut-il appliquer sur les entrées J et K pour l'obtenir ? (fig. 10.17).

Ainsi la ligne 1 de la table de transition s'obtient à partir des lignes 4 et 6 du tableau de vérité.

$QQ^+ = 01$ s'obtient par $J = 1$, $K = 0$ (cas 4) ou par $JK = 11$ (cas 6) soit $J = 1$ et $K = 0$ ou 1 soit $JK = 1X$.

Cette table de transition sert au calcul des fonctions J et K à appliquer pour des bascules devant décrire des séquences imposées en sortie.

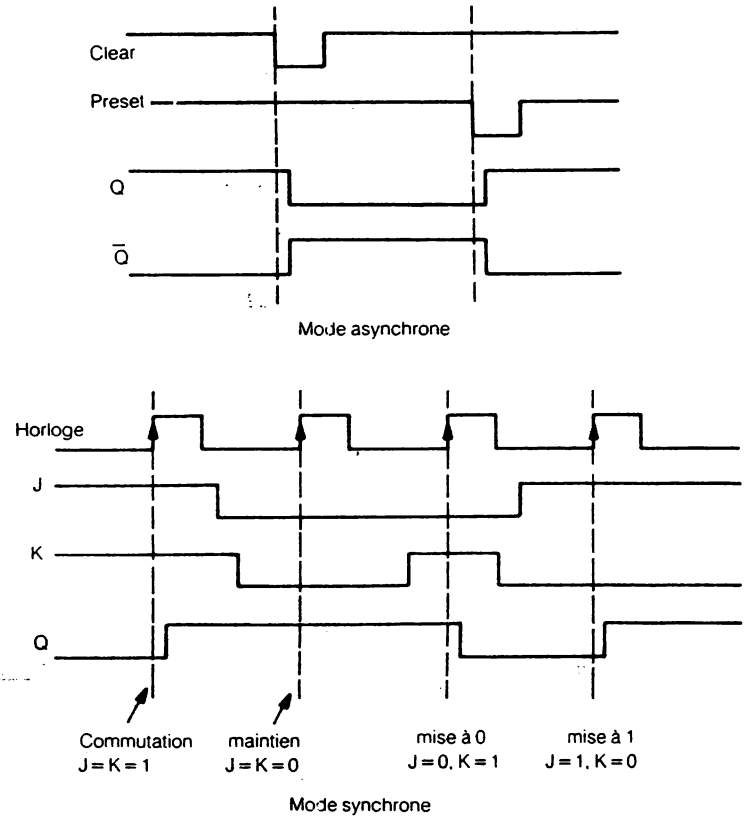


Fig. 10.15. — *Diagramme des temps d'une bascule JK.*

Cas	J	K	Q	Q^+
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

Fig. 10.16. — *Table de vérité complète d'une bascule JK.*

Cas	Q	Q^+	J	K
0	0	0	0	X
1	0	1	1	X
2	1	0	X	1
3	1	1	X	0

Fig. 10.17. — *Table de transition d'une bascule JK.*

7.5. Équation d'une bascule JK

On utilise quelquefois l'équation donnant la sortie Q^+ en fonction de J, K, Q. On l'obtient à partir de la table de vérité de la figure 10.16 et après simplification (fig. 10.18).

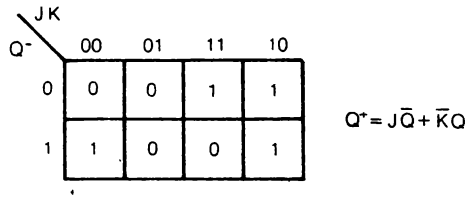


Fig. 10.18. — *Simplification de la fonction Q^+.*

7.6. Exemple de bascule JK

7.6.1. Brochage du boîtier 74 LS 112 A

Le boîtier 74 LS 112 A, à 16 broches, contient deux bascules JK, ayant chacune des commandes Preset, Clear, J, K, Clock indépendantes (fig. 10.19).

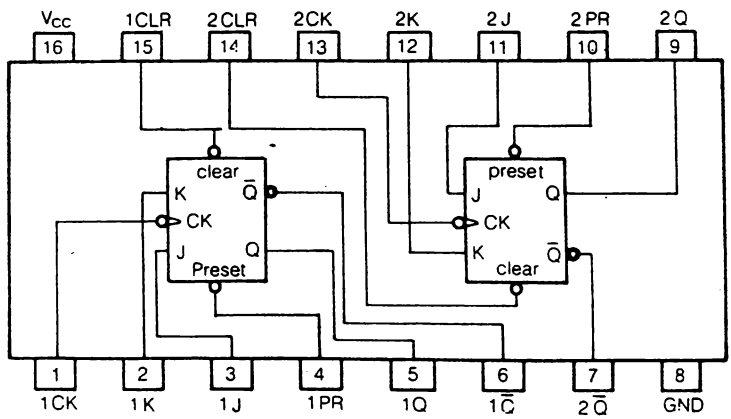


Fig. 10.19. — *Brochage du boîtier 74 LS 112 A.*

7.6.2. Tableau de vérité

La bascule *JK* 74 LS 112 A fonctionne sur un front de descente de l'horloge. Son tableau de vérité se déduit de celui de la figure 10.14 en remplaçant la flèche montante par une flèche descendante.

7.6.3. Paramètres électriques (fig. 10.20)

La plupart des paramètres sont identiques à ceux de la porte standard 74 LS 00 sauf l'entrance des entrées Clear, Preset et Clock. Clear et Preset comptent pour trois charges au niveau haut et deux charges au niveau bas, tandis que Clock compte pour quatre charges au niveau haut et deux charges au niveau bas. Dans la pratique, on compte le maximum de charges pour une entrée donnée.

7.6.4. Paramètres dynamiques

La figure 10.21 donne les paramètres dynamiques de la bascule 74 LS 112 A. Cette bascule fonctionne de manière garantie avec une fréquence d'horloge comprise entre 0 et 30 MHz. La seconde ligne du tableau indique que d'une manière générale, les bascules pourront fonctionner à 35 MHz voire plus, mais le constructeur ne garantit que la plage 0 à 30 MHz. Il est donc plus prudent de rester dans la plage 0-30 MHz. La figure 10.21 donne aussi la durée minimum des impulsions à appliquer sur les entrées Clear, Preset et Clock. Le temps minimum de préaffichage est de 20 *ns* par rapport au front de descente tandis que le temps de maintien est de 0 *ns* par rapport au même front (voir figures 5.15 et 10.22 pour la définition de ces deux paramètres). Enfin on indique le temps de propagation des différentes entrées vers les sorties Q et $\bar{Q}$, spécifié en valeurs typique et maximum, mais pas en valeur minimum. La bascule 74 LS 112 A est du type edge triggered (déclenchement par front d'horloge) ce que l'on voit dans le tableau des paramètres dynamiques ($t_{set\ up}$ et t_{hold} référencés par rapport au même front). Par contre, si nous avions étudié la bascule 7473, nous aurions trouvé un $t_{set\ up}$ de 0 *ns* par rapport au front de montée de l'horloge et un t_{hold} de 0 *ns* par rapport au front de descente de l'horloge. Ceci implique que les données J et K restent stables pendant toute la durée de l'horloge. La

Paramètres		NOM MIN	NOM TYP	NOM MAX	Unité	COMPARAISON AVEC LA PORTE 74 LS 00
Tension d'alimentation V_{CC}		4,75	5	5,25	V	Idem
Courant de sortie, niveau haut I_{OH}				−400	µA	Idem
Courant de sortie niveau bas I_{OL}				8	mA	Idem
Tension d'entrée niveau haut V_{IH}		2			V	Idem
Tension d'entrée niveau bas V_{IL}				0,8	V	Idem
Tension d'écrêtage V_{IK}				−1,5	V	Idem
Tension de sortie niveau haut V_{OH}		2,7	3,4		V	Idem
Tension de sortie niveau bas V_{OL}			0,35	0,5	V	0,25 et 0,5 respectivement
Courant d'entrée I_I à la tension maximum d'entrée $V_I = 7$ V	J K			0,1	mA	Idem
	CLR, PR			0,3	mA	0,1 mA
	CK			0,4	mA	0,1 mA
Courant d'entrée, niveau haut I_{IH}	J, K			20	µA	Idem
	CLR, PR			60	µA	20 µA
	CK			80	µA	20 µA
Courant d'entrée, niveau bas I_{IL}	J, K			−0,4	mA	Idem
	CLR, PR			−0,8	mA	− 0,4 mA
	CK			−0,8	mA	− 0,4 mA
Courant de court-circuit I_{OS}		− 20		− 100	mA	Idem
Courant d'alimentation I_{CC}			4	6	mA	0,8 1,6 (sorties hautes) 2,4 4,4 (sorties basses)

Fig. 10.20. — *Paramètres électriques de la bascule JK 74 LS 112 A.*

bascule 7473 est ainsi du type pulse triggered (déclenchement par impulsion).

Le diagramme des temps de la figure 10.22 illustre la plupart des différents paramètres temporels de la bascule 74 LS 112 A.

Paramètres		NOM MIN	TYP	MAX	Unité
Fréquence de l'horloge f_c		0		30	MHz
Fréquence maximum d'horloge f_c		30	35 *		MHz
Durée d'impulsion t_w	Horloge haute	20			nS
	Preset et Clear bas	25			nS
Temps de préaffichage (set up) $t_{set\ up}$	Donnée niveau haut	20 ↓			nS
	Donnée niveau bas	20 ↓			nS
Temps de maintien (hold time) t_{hold}		0 ↓			nS
Temps de propagation du niveau bas au niveau haut t_{PLH}, pour Preset, Clear, Clock vers Q, $\overline{Q}$			15	20	nS
Temps de propagation du niveau haut au niveau bas t_{PHL}, pour Preset, Clear, Clock vers Q, $\overline{Q}$			15	20	nS

* Cette fréquence n'est pas garantie par le constructeur

Fig. 10.21. — *Paramètres dynamiques de la bascule 74 LS 112 A.*

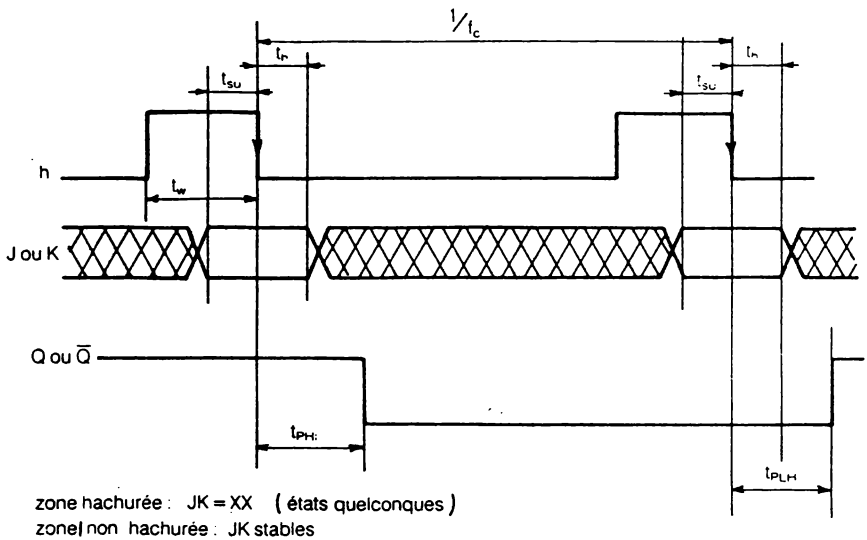


Fig. 10.22. — *Paramètres temporels de la bascule 74 LS 112 A.*

7.6.5. Variété des bascules JK

A la lecture d'un catalogue de circuits intégrés, la profusion des bascules *JK* surprend :

- Bascules "pulse triggered" (déclenchement par impulsion).
- Bascules "negative edge triggered" (déclenchement sur front descendant).
- Bascule " positive edge triggered" (déclenchement sur front montant).
- Entrées *JK* précédées de porte ET et de portes ET-OU.
- Bascules "data lock-out" (vérouillage des données).
- Bascules avec certaines entrées communes (Clear, Preset, Clock) ou séparées.

Pourquoi toutes ces bascules ? Parce que les bascules *JK* ont été les premières bascules introduites par les constructeurs et qu'entre 1965-1968, les concepteurs de système n'avaient pour tout outil logique que les portes et les bascules *JK*. D'où pour répondre à des besoins très variés, une grande diversité de bascules.

Aujourd'hui, tout cela n'a plus la même importance, les concepteurs disposant de circuits complexes variés intégrant des sous-ensembles de bascules. La bascule *JK* n'occupe plus alors qu'une place marginale dans la conception. Un seul type de bascule de type "edge triggered" peut suffire à l'ensemble des besoins.

8. LA BASCULE D

8.1. Schéma symbolique

La figure 10.23 donne le schéma symbolique d'une bascule *D*.

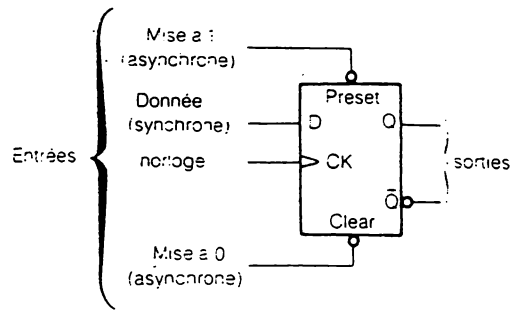


Fig. 10.23. – *Schéma symbolique de la bascule D.*

La bascule D comprend :

- Une entrée Preset de mise à 1 asynchrone, inverse.
- Une entrée Clear de mise à 0 asynchrone, inverse.
- Une entrée horloge CK, qui valide l'entrée D, active sur un front montant.
- Une entrée D (donnée) synchrone.
- Deux sorties complémentaires Q et $\bar{Q}$.

8.2. Fonctionnement statique

La table de vérité de la bascule D (fig. 10.24) décrit les fonctionnements synchrone et asynchrone. Le mode asynchrone est identique à celui de la bascule JK. Le mode synchrone est très simple.

- Au front montant d'horloge, la donnée présente sur D est transférée en sortie.
- En l'absence de front montant d'horloge, l'état de la bascule se conserve.

	Cas	Preset	Clear	Clock	D	Q^-	$\bar{Q}^+$	Remarques
Mode asyn.	0	0	1	X	X	1	0	mise à 1
	1	1	0	X	X	0	1	mise à 0
	2	0	0	X	X	1	1	Etat instable
Mode synch.	3	1	1	↑	0	0	1	Mise à 0
	4	1	1	↑	1	1	0	Mise à 1
	5	1	1	0 ou 1	X	Q	$\bar{Q}$	Maintien

Fig. 10.24. — *Tableau de vérité de la bascule D.*

8.3. Diagramme de fonctionnement

Ce diagramme est très simple. La sortie Q recopie l'entrée D (fig. 10.25).

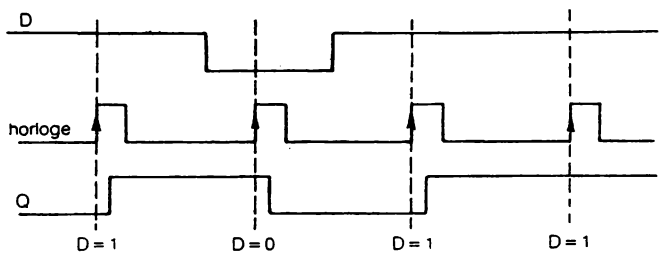


Fig. 10.25. — *Diagramme des temps d'une bascule D.*

8.4. Équation et table de transition de la bascule D

L'équation est également très simple: $Q^+ = D$; et la table de transition aussi car l'état futur ne dépend pas de l'état Q (comme dans une bascule JK) mais seulement de D (fig. 10.26).

Q^+	D
0	0
1	1

Fig. 10.26. – *Table de transition d'une bascule D.*

8.5. Exemple de bascule D

Nous prenons à titre d'exemple le boîtier 74 LS 74 A à 14 broches. Il contient deux bascules D indépendantes ayant le schéma symbolique de la figure 10.23. Il s'agit d'une bascule déclenchée par front montant (positive edge triggered). Son tableau de vérité correspond à celui de la figure 10.24. Comme pour la bascule JK, certaines entrées présentent des entrances supérieures à la normale: deux charges pour Preset et Clear. La figure 10.27 donne les paramètres dynamique de cette bascule.

Paramètres		MIN	NOM TYP	MAX	Unité
Fréquence de l'horloge		0		25	MHz
Fréquence maximum d'horloge		25	33 *		MHz
Durée d'impulsion t_w	Horloge haute	25			nS
	Preset et Clear bas	25			nS
Temps de préaffichage $t_{set\ up}$	Donnée niveau haut	25 ↑			nS
	Donnée niveau bas	20 ↑			nS
Temps de maintien t_{hold}		5 ↑			nS
Temps de propagation t_{PLH} du niveau bas au niveau haut, des entrées Preset, Clear ou Clock vers Q ou $\bar{Q}$.			13	25	nS
Temps de propagation t_{PHL} du niveau bas au niveau haut des entrées Preset, Clear ou Clock vers Q ou $\bar{Q}$.			25	40	nS

* Cette fréquence n'est pas garantie par le constructeur

Fig. 10.27. – *Paramètres dynamiques de la bascule D 74 LS 74 A.*

9. LE MONOSTABLE

9.1. Schéma symbolique

La figure 10.28 montre le schéma symbolique d'un monostable de type 74 LS 123. Il comprend deux entrées de déclenchement A et B, une entrée d'initialisation Clear ou mise à zéro et deux sorties inverses l'une de l'autre Q et $\bar{Q}$. Le réseau RC qui détermine la durée de l'impulsion recueillie en sortie Q et $\bar{Q}$ se branche sur les bornes R_{ext} et C_{ext}. Le boîtier 74 LS 123 contient deux monostables.

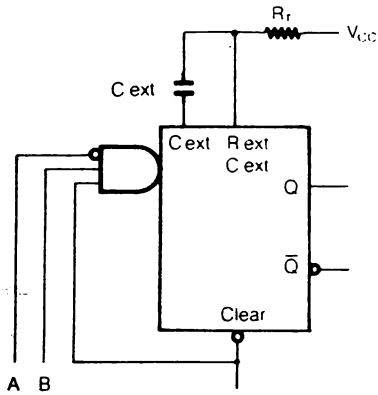


Fig. 10.28. — *Schéma symbolique du monostable 74 LS 123.*

Entrees			Sorties	
Clear	A	B	Q	Q̄
0	X	X	0	1
X	1	X	0	1
X	X	0	0	1
1	0	↑	⎍	⎎
1	↓	1	⎍	⎎
↑	0	1	⎍	⎎

Fig. 10.29. — *Table de vérité du monostable 74 LS 123.*

9.2. Fonctionnement

Le tableau de vérité de la figure 10.29 s'interprète ainsi :

— Si l'entrée Clear est à 0, quelles que soient les entrées A et B, les sorties Q et $\bar{Q}$ sont dans leur état inactif ($Q = 0$, $\bar{Q} = 1$).

— Si l'entrée A est à 1, quelles que soient les entrées Clear et B, les sorties Q et $\bar{Q}$ sont dans leur état inactif ($Q = 0$, $\bar{Q} = 1$).

— Si l'entrée B est à 0, quelles que soient les entrées A et Clear, les sorties Q et $\bar{Q}$ sont dans leur état inactif ($Q = 0$, $\bar{Q} = 1$).

– Si Clear est à 1, A à 0, et si une transition positive se produit sur B, la sortie Q passe en 1 pour une durée T et la sortie $\overline{Q}$ passe à 0 pour la même durée T.

– Si Clear est à 1, B à 1, et si une transition négative se produit sur A, les sorties Q et $\overline{Q}$ sont déclenchées pour une durée T.

– Si A est à 0, B à 1, et si une transition positive se produit sur Clear, les sorties Q et $\overline{Q}$ sont déclenchées pour une durée T.

9.3. Diagramme des temps de fonctionnement

La figure 10.30 montre quelques cas de fonctionnement de ce monostable : déclenchement sur un front de descente de A ou un front de montée de B, déclenchement puis arrêt par une impulsion Clear, déclenchement et redéclenchement par une seconde impulsion d'entrée avant que l'impulsion de sortie ne soit terminée.

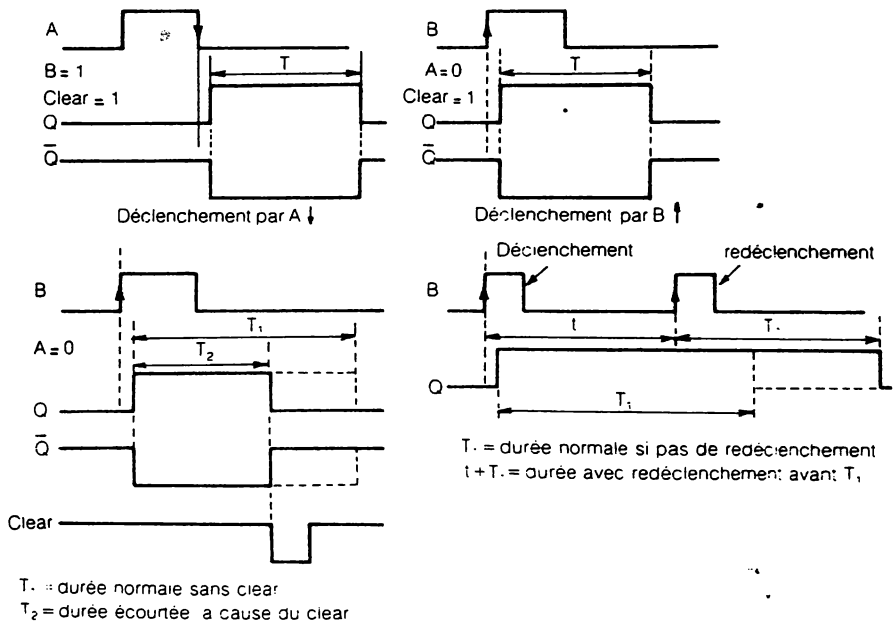


Fig. 10.30. — *Fonctionnements d'un monostable.*

9.4. Domaine d'emploi

Le monostable sert à créer un retard donné, à élaborer une suite d'impulsions dans le temps. Comme l'impulsion d'entrée et de sortie peuvent être de durées très différentes, le monostable sert d'interface entre des systèmes lents et rapides, par exemple pour commander un relais électromagnétique réagissant en 10 *ms* avec des impulsions de quelques microsecondes, trop brèves pour exciter directement le relais.

Tout comme les bascules, le monostable est caractérisé par des paramètres électriques et des paramètres dynamiques. Parmi ces derniers figurent notamment la durée minimum de l'impulsion de déclenchement, les retards au déclenchement et des formules ou des abaques donnant la durée de l'impulsion de sortie en fonction du réseau RC employé.

10. L'ASTABLE OU OSCILLATEUR

L'oscillateur fournit un signal périodique, de durée d et de période T (fig. 10.31). Ce signal sert d'horloge aux circuits séquentiels. Le rapport $\frac{d}{T}$ s'appelle rapport cyclique (duty cycle). On peut réaliser des oscillateurs de différentes manières:

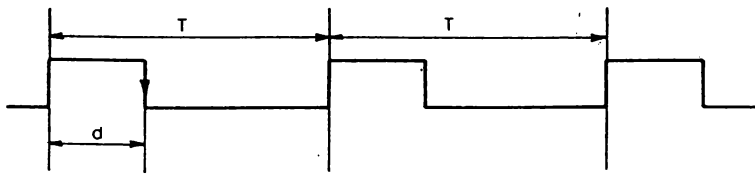


Fig. 10.31. — *Cycle d'horloge fourni par un oscillateur.*

— A partir de trigger de Schmitt, en utilisant la décharge et la charge d'un condensateur entre les deux seuils de basculement V_T- et V_T+ (fig. 10.32).

— A partir de deux monostables, la sortie du premier excitant un second, le second excitant le premier.

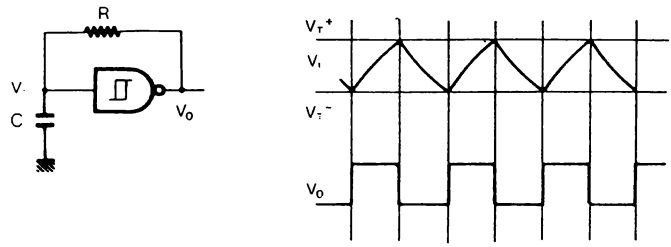


Fig. 10.32. — *Oscillateur à trigger de Schmitt.*

— A partir d'une chaîne d'inverseurs rebouclés avec un quartz dans la chaîne (fig. 10.33).

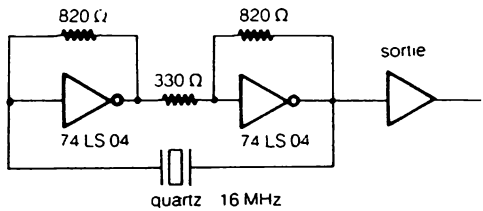


Fig. 10.33. — *Oscillateur 16 MHz à quartz et composants discrets.*

— En utilisant des boîtiers oscillateurs, prévus à cet usage, comme le 74 LS 326 (fig. 10.34). La valeur de la capacité branchée à l'extérieur du circuit et le niveau de tension, compris entre 0 *V* et 5 *V*, appliqué à l'entrée "frequency control" déterminent la valeur de la fréquence d'oscillation. Un signal carré de rapport cyclique 50 % apparaît en sortie *Y* et $\overline{Y}$ si l'entrée "Enable" est à 0.

Pour obtenir une période plus stable, on peut brancher un quartz à la place de la capacité. On obtient alors une dérive très faible : 10^{-6} en valeur relative, soit quelques dizaines de secondes par an. La plupart des applications s'accomodent d'une telle dérive.

Les montages utilisant des capacités et des résistances présentent une fréquence d'oscillation peu précise, car dépendant des paramètres des circuits, des valeurs des composants *R* et *C*. Il faut ajuster les valeurs des composants pour obtenir la fréquence désirée. En outre, elle varie dans d'assez larges proportions avec la température et la tension d'alimentation. Les montagnes avec monostables ont des risques de blocage (plus d'oscillation). En conclusion, il vaut mieux se servir d'un boîtier oscillateur prêt

à l'emploi si l'on ne maîtrise pas parfaitement la technique des circuits analogiques.

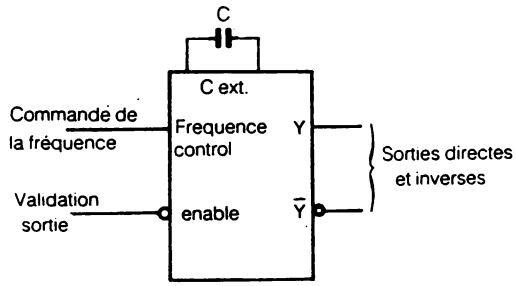


Fig. 10.34. — *Oscillateur à fréquence asservie 74 LS 326.*

Annexe

BISTABLES ET BASCULES PRINCIPAUX CIRCUITS INTÉGRÉS

1. LES BISTABLES

Les bistables sont des éléments de mémorisation simple qui ne permettent pas des automodifications et des réactions croisées (voir chapitre 10, paragraphe 6).

Il existe assez peu de bistables intégrés. La figure 10.35 en donne quatre types : un boîtier de quatre bistables $\overline{RS}$ indépendants et trois boîtiers de quatre bistables D chacun, avec des commandes communes.

Les boîtiers de bistables D ont donc plutôt une orientation registre, les bistables n'étant pas complètement indépendants.

Le tableau de la figure 10.35 donne les caractéristiques principales de ces bistables.

Référence	Nombre broches	Nombre opérateurs	Type	Sorties	Horloge	Signaux de commande min en ns			Temps de propagation	observations
						t_w	t_{su}	t_h	t_c max	
74 LS 279	16	4	$\overline{R}\,\overline{S}$	Q	–	27	–	–	27	sans preset ni
74 LS 75	16	4	D	$Q\,\overline{Q}$	⎍	20	20	0	19	clear – bistables
74 LS 77	14	4	D	Q	⎍	20	20	0	19	associés par paires
74 LS 375	16	4	D	$Q\,\overline{Q}$	⎍	20	20	0	19	(horloge commune)

t_w : Durée de la commande

t_{su} : Temps de préaffichage

t_h : Temps de maintien

t_p : Temps de propagation le plus défavorable (t_{PLH} ou t_{PHL})

⎍ : action sur niveau haut

Fig. 10.35. — *Les bistables intégrés.*

Dans ce tableau seules les valeurs max les plus défavorables des délais de propagation ont été reportés quelque soit l'origine de la commande ($\overline{R}$, $\overline{S}$, D, horloge).

Les trois boîtiers de bistables D diffèrent très peu les uns des autres, les 74 LS 75 et 74 LS 77 se caractérisent par un brochage d'alimentation (5 V et masse) non normalisé, le 74 LS 375 est une variante du 74 LS 75 à brochage normalisé.

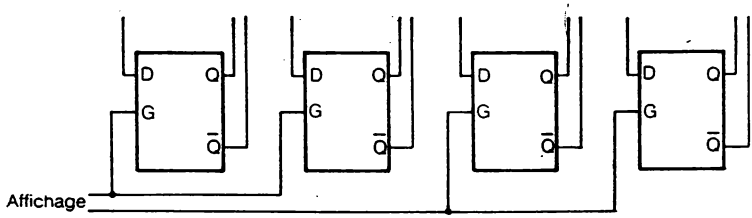


Fig. 10.36. — *Schéma de principe des bistables 74 LS 75, 74 LS 77, 74 LS 375.*

La figure 10.36 donne le schéma de principe commun à ces bistables ; dans le cas du 74 LS 77 seules les sorties Q sont accessibles.

2. LES BASCULES

Les bascules sont des éléments de mémorisation plus complexes que les bistables qui permettent les automodifications et les réactions croisées (voir chapitre 10, § 6). On dénombre neuf types de boîtiers de bascules de technologie *LS* dans un catalogue de constructeur (fig. 10.37).

Référence	Nombre broches	Nombre opérateurs	Type	Sorties	horloge	Preset	Clear	Signaux de commande min en ns : t_w horloge	t_{su}	t_h	propagation max en ns : t_p	observations
74 LS 73A	14	2	JK	$Q\bar{Q}$	↓	–	I	20	20	0	20	brochage non normalisé
74 LS 107	14	2	JK	$Q\bar{Q}$	↓	–	I	20	20	0	20	
74 LS 113	14	2	JK	$Q\bar{Q}$	↓	I	–	20	20	0	20	
74 LS 76A	16	2	JK	$Q\bar{Q}$	↓	I	I	17	20	0	20	brochage non normalisé
74 LS 112	16	2	JK	$Q\bar{Q}$	↓	I	I	17	20	0	20	
74 LS 78A	14	2	JK	$Q\bar{Q}$	↓	I	I	17	20	0	20	Clear commun différent par le brochage
74 LS 114	14	2	JK	$Q\bar{Q}$	↓	I	I	17	20	0	20	
74 LS 109	16	2	$J\bar{K}$	$Q\bar{Q}$	↑	I	I	20	20	5	40	
74 LS 74A	14	2	D	$Q\bar{Q}$	↑	I	I	20	25	5	40	

↓ Action de l'horloge sur front de descente

↑ Action de l'horloge sur front de montée

t_{su} et t_h référencé par rapport au front actif de l'horloge

I Action de la commande par un zéro logique (masse)

Fig. 10.37. — *Les bascules intégrées.*

Sept types *JK* peu différents les uns des autres que l'on peut cependant classer en trois sous familles :

- 74 LS 73A, 74 LS 107, 74 LS 113 qui diffèrent uniquement par la présence ou non de commande clear et preset et par des brochages différents.
- 74 LS 76A, 74 LS 112 variante des *JK* précédents possédant les deux commandes clear et preset en deux brochages différents.
- 74 LS 78A, 74 LS 114 autres variantes avec clear commun aux deux bascules intégrées dans un boitier en deux brochages différents.

La coexistence au catalogue d'un constructeur de 9 boîtiers aussi peu différents s'explique par l'histoire. Un seul boîtier, le 74 LS 112 par exemple,pourrait répondre à tous les besoins, mais le constructeur ne peut retirer les autres du catalogue car ils restent utilisés dans des applications diverses et leur suppression entraînerait des reprises de circuits imprimés (brochages différents) que la clientèle ne peut accepter. Le 74 LS 109 est un boîtier un peu spécial, $J\overline{K}$ réagissant sur front montant d'horloge. Enfin il existe une double bascule type D (74 LS 74A) possédant des caractéristiques externes proches de la bascule $J\overline{K}$ précédente.

Chapitre 11

LA FONCTION DE MÉMORISATION

1. LES FONCTIONS SÉQUENTIELLES COMPLEXES USUELLES

Le chapitre précédent présentait les fonctions séquentielles simples, n'intéressant qu'un seul élément d'information. Nous abordons dans ce chapitre et les suivants, l'étude des fonctions séquentielles usuelles portant sur plusieurs éléments d'information à la fois. Ce regroupement porte le nom de registre. La figure 11.1 donne la structure la plus générale d'un tel circuit.

Il comprend :

- Un groupe de bascules, alimentées par une horloge.
- Un réseau combinatoire d'entrée, qui, à partir de commandes d'entrée, d'informations d'entrée et de l'état des bascules élabore les commandes d'excitation des bascules.
- Un réseau combinatoire de sortie, généralement très simple, qui, à partir de l'état des bascules et des commandes de sortie donne l'état de sortie du registre.

Le schéma bloc de la figure 11.1 correspond au cas le plus complexe. Certains registres ont une structure plus simple avec un seul bloc combinatoire ou même aucun bloc combinatoire. Les registres possèdent des

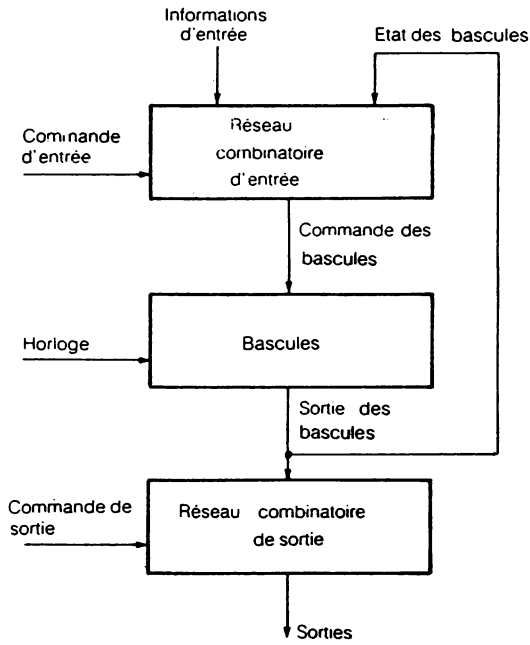


Fig. 11.1. — *Structure générale d'un registre.*

commandes communes qui agissent sur l'ensemble des éléments binaires, ce qui les distingue d'un groupe de bascules indépendantes.

Le fonctionnement d'un registre se décrit par un tableau de vérité (fig. 11.2) où figurent en entrée :

- Les états anciens des bascules ;
- Les informations d'entrée ;
- Les commandes d'entrée ;
- Les commandes de sorties ;
- L'horloge ;

pour fournir en sortie :

- Les états futurs des bascules ;
- Les états des sorties du registre.

Comme pour les bascules, des tableaux de paramètres électriques et temporels précisent les niveaux, la durée, la position des signaux dans le temps, etc.

Entrées					Sorties	
Etat ancien des bascules	Informations d'entrée	Commande d'entrée	Commande de sorties	Horloge	Etat futur des bascules	Etat des sorties
$Q_{n-1} \cdots Q_1 Q_0$					$Q_{n-1} \cdots Q_1 Q_0$	$Y_{n-1} \cdots Y_1 Y_0$

Fig. 11.2. — *Forme d'un tableau de vérité de registre.*

Des relations a priori quelconques peuvent être définies par le tableau de vérité de la figure 11.2. En pratique, seuls trois types de fonctionnement ont été retenus :

- La fonction de mémorisation : une information placée aux entrées du registre est copiée dans celui-ci.
- La fonction de comptage : si le registre contient une valeur ancienne n, il affiche ensuite une valeur future $n + 1$. On trouve également la fonction de décomptage.
- La fonction de décalage : l'état des bascules glisse d'un rang, Q_i prend l'état ancien de Q_{i-1}, qui prend l'état ancien de Q_{i-2}, etc... Il existe des décalages dans les deux sens (droite ou gauche).

On trouve ainsi dans les catalogues des registres de mémorisation, de comptage, et à décalage. Dans ce chapitre, nous nous intéressons à la fonction de mémorisation.

Historiquement, les premiers registres de mémorisation ont été conçus avec des bistables *RS*. Aujourd'hui, on emploie essentiellement les bistables *D* et les bascules *D*.

On ne les réalise plus avec des éléments individuels, les constructeurs fournissant des boîtiers intégrant plusieurs éléments binaires. Les registres de mémorisation intégrés sont nombreux. Aussi, afin d'aider les lecteurs dans l'exploitation des catalogues de constructeurs, nous avons dégagé quelques caractéristiques principales, telles que commandes communes, modes d'affichage, modes d'exploitation des sorties. Ces notions sont valables également pour les registres de comptage et de décalage étudiés dans les chapitres suivants.

2. LES COMMANDES COMMUNES D'UN REGISTRE

2.1. Principe

La figure 11.3 montre le schéma de principe d'un registre de mémorisation, dont la structure est plus simple que celle du registre de la figure 11.1. Les quatre informations d'entrées sont copiées simultanément dans les quatre bascules *D* sous l'influence du signal d'horloge. Les quatre bascules peuvent être mises à zéro simultanément et les sorties du registre sont validées pour donner les quatre informations de sortie. Toutes les commandes citées agissent globalement sur le registre : une action individuelle sur une bascule n'est pas possible.

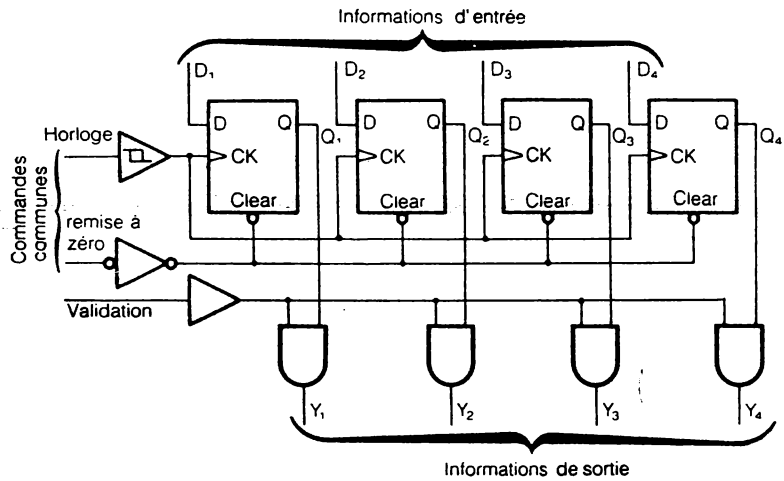


Fig. 11.3. — *Commandes communes dans un registre.*

2.2. Charge présentée par une commande

Afin de réduire l'entrance des commandes communes qui desservent plusieurs entrées, un inverseur ou un amplificateur est placé entre la commande et les entrées correspondantes des bascules. A la figure 11.3, l'entrance de l'horloge est de une charge. Sans amplification, elle aurait été de quatre charges avec des bascules *D* 74 LS 74A. Ceci facilite la constitution de grands registres par juxtaposation de registres élémentaires.

2.3. Priorité entre commandes

Toutes les commandes n'ont pas la même priorité, cela signifie que si l'on applique deux commandes simultanées plus ou moins contradictoires, le circuit qui ne peut occuper qu'un état donné suivra une des commandes en ignorant l'autre. Dans le registre de la figure 11.3, la remise à zéro est prioritaire par rapport à la commande d'affichage, définie par l'horloge. Cela se traduit dans le tableau de vérité de la figure 11.4.

Entrées			Sorties
Remise à 0	Information D_i	Horloge	Q_i^+
0	X	X	0
1	D_i	↑	D_i
1	X	sans transition	Q_i

La mise à 0 l'emporte quelle que soit la commande d'affichage

L'affichage n'est effectif que si la remise à zéro est inactive

Fig. 11.4. — *Priorité entre commandes.*

2.4. Commande avec trigger de Schmitt

Afin de réduire leur sensibilité aux bruits électriques extérieurs, les commandes communes disposent parfois d'entrée de type trigger de Schmitt (cas de l'entrée horloge à la figure 11.3).

3. LES MODES D'AFFICHAGE

On entend par mode d'affichage, la nature de la commande mémorisant l'information d'entrée du registre. On distingue la commande par état ou par impulsion (pulse triggered) et la commande par front (edge triggered).

3.1. Registres à commande d'affichage par état

Ils sont réalisés avec des bistables *D* présentés au chapitre 10. La figure 11.5 montre le schéma de principe et la figure 11.6, la table de vérité.

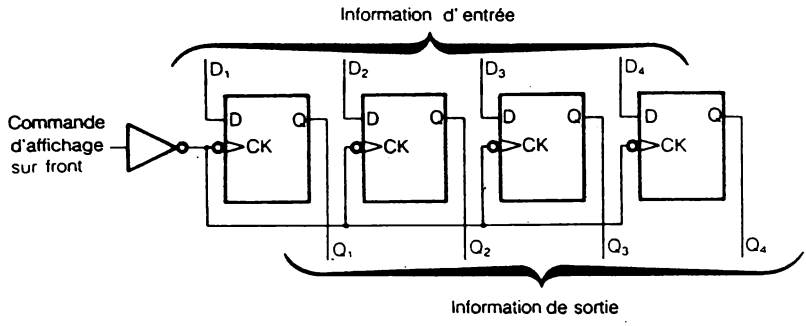


Fig. 11.5. — *Registre à commande d'affichage par état. Schéma de principe.*

Entrées		Sorties	
D_i	G_i	Q_i^+	
0	1	0	Transfert
1	1	1	
X	0	Q_i	Mémorisation

X état indifférent
Q_i état mémorisé

Fig. 11.6. — *Registre à commande d'affichage par état. Table de vérité.*

La table de vérité indique deux modes de fonctionnement :

— La mémorisation quand $G = 0$. Le registre conserve son état antérieur. Il est insensible aux données présentées sur D.

— Le transfert, quand $G = 1$. La sortie du registre prend la valeur présentée sur D. Si pendant que $G = 1$, l'entrée D varie, la sortie Q suit la variation de D. On dit alors que le registre est *transparent.*

La figure 11.7 montre le fonctionnement d'un tel registre. On note que la sortie recopie l'entrée avec un temps de retard, sur G quand D est stable ou avec un temps de retard sur D quand G est stable.

Les concepteurs de systèmes logiques utilisent fréquemment la propriété de transparence pour exploiter très rapidement une information (dès que $G = 1$) et qui sera mémorisée ultérieurement (quand G revient à 0).

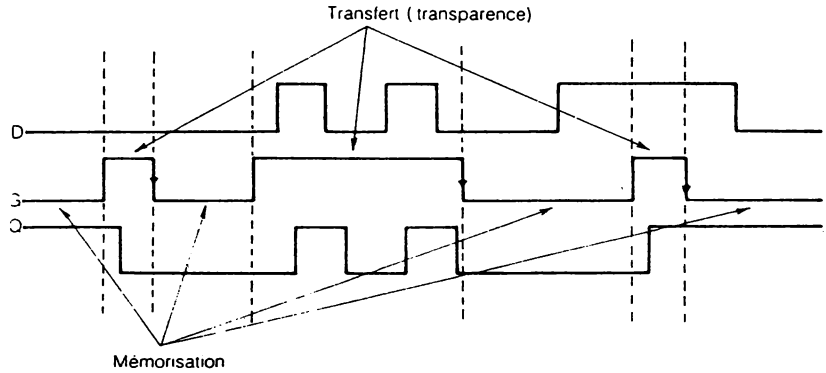


Fig. 11.7. – *Fonctionnement d'un registre latch D.*

3.2. Registres à affichage sur front d'horloge

3.2.1. Principe

Ils sont réalisés à partir de bascules *D*, mémorisant l'information au voisinage d'un front d'horloge montant ou descendant. La figure 11.8 donne le schéma de principe d'un tel registre et la figure 11.9, la table de vérité.

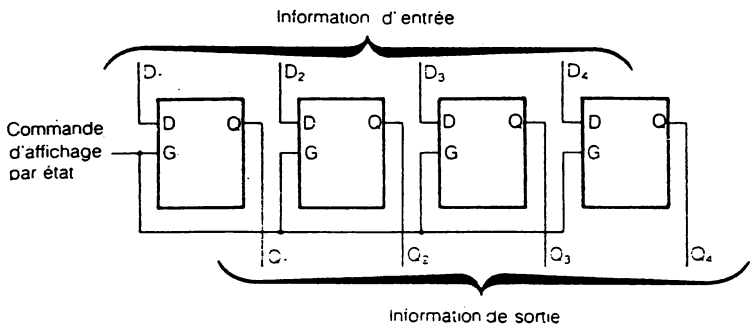


Fig. 11.8. – *Registre à affichage sur front d'horloge.*

Entrées		Sorties
Horloge	D	Q^-
sans transition	X	Q
↑	1	1
↑	0	0

X : Etat indifférent
↑ Transition du niveau bas vers le niveau haut du signal d'horloge
Q^- Etat mémorisé sur la transition précédente

Fig. 11.9. – *Affichage sur front d'horloge. Table de vérité.*

3.2.2. Affichage d'une information dans un registre à commande sur front d'horloge

La plupart du temps, une information est chargée dans un registre R par l'impulsion d'horloge h en tenant compte de conditions logiques C suivant l'équation logique :

$C.(h\uparrow) \rightarrow$ Données dans registre R

Pour traduire cette relation au niveau du schéma, on présente le produit $C.h$ sur la commande horloge et les données sur les entrées D (fig. 11.10). Mais en effectuant ce montage, nous transformons la relation $C.(h\uparrow)$ en $(C.h)\uparrow$ qui n'est pas toujours équivalente comme le montre la figure 11.11.

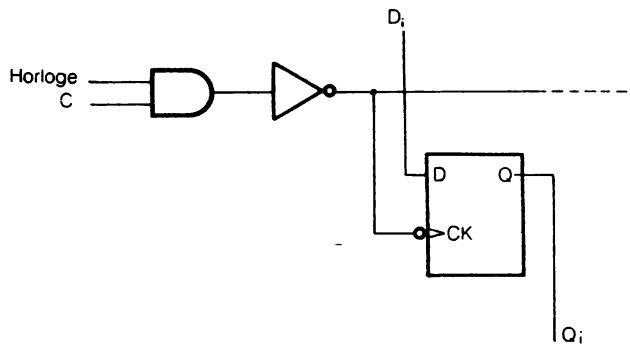


Fig. 11.10. — *Affichage conditionnel d'un registre.*

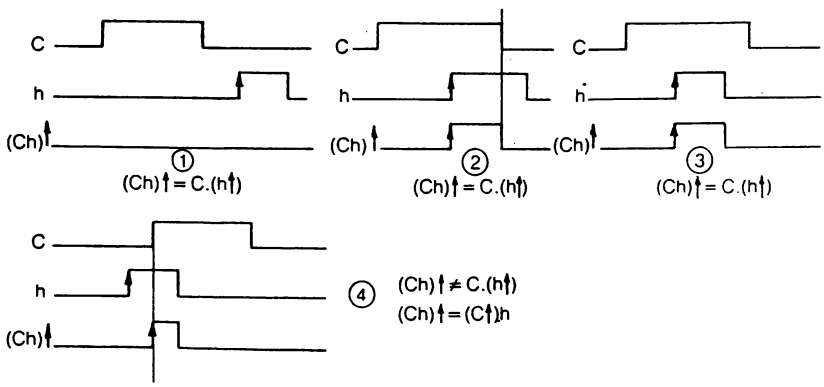


Fig. 11.11. — *Les formes possibles du produit (Ch)↑.*

En particulier, si la condition C provient de l'état d'un registre qui se modifie au moment du front montant de h (automodification par exemple), elle sera instable après le front montant de h. On aura donc des mémorisations décalées erronées (fig. 11.12).

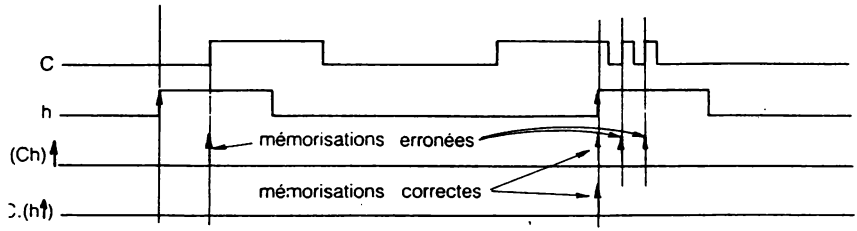


Fig. 11.12. — *Influence de la commande sur la mémorisation.*

Afin d'éviter l'inconvénient précédent, les constructeurs ont imaginé des registres avec validation d'affichage et horloge indépendante permettant de respecter la condition $C.(h\uparrow)$.

3.2.3. Registre avec validation d'affichage et horloge indépendante

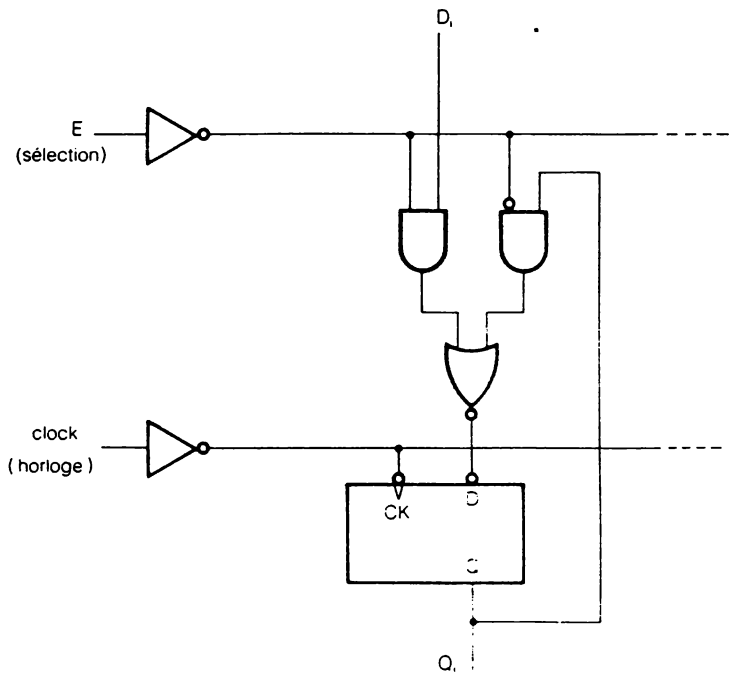


Fig. 11.13. — *Cellule de registre avec validation d'affichage.*

Chaque cellule du registre se compose d'une bascule D et d'un multiplexeur de type 2 vers 1 (fig. 11.13). Quand E vaut 0, le multiplexeur sélectionne les entrées D qui sont copiées dans les bascules par l'horloge. Quand E vaut 1, le multiplexeur sélectionne les sorties du registre de sorte que lorsque l'horloge se présente, elle recopie le contenu du registre dans lui-même. La table de vérité de la figure 11.14 résume le fonctionnement de cette cellule.

Entrées			Sorties
E	D_i	CK	Q_i
X	X	sans transition	$\overline{Q_i}$
0	D_i	↑	D_i
1	X	↑	$\overline{Q_i}$

X : Etat logique indifférent
↑ : Front montant de l'horloge
$\overline{Q_i}$: Etat précédent de Q_i

Fig. 11.14. — *Registre avec validation d'affichage. Table de vérité.*

Avec une telle cellule, l'équation $C.(h\uparrow)$ peut être parfaitement respectée en présentant h sur l'entrée horloge, la condition C sur l'entrée E et la donnée à mémoriser sur l'entrée D_i. Il suffit pour que le fonctionnement soit convenable que la condition appliquée en E et les données D_i respectent les conditions de temps de préaffichage (t_{su}) et de maintien (t_{hold}) imposées par le constructeur. Des variations de C pendant h, au-delà du temps de maintien n'affectent pas la mémorisation.

3.2.4. Registre avec sélection des données en entrée

Quelques registres disposent d'une possibilité de multiplexage au niveau de l'affichage des données, ainsi que le montre le schéma de la figure 11.15 et la table de vérité de la figure 11.16.

En pratique, les registres sont du type RS synchrone pour l'utilisateur ; leur mode d'exploitation est analogue à celui de registres réalisés avec des bascules D ainsi que le montre la table de vérité de la figure 11.16.

L'entrée de sélection permet d'effectuer le choix de l'information D_{1b} ou D_{1a} à afficher dans le registre, évidemment les conditions de préaffichage t_{su} et de maintien t_h par rapport au front montant de l'horloge devront être respectées.

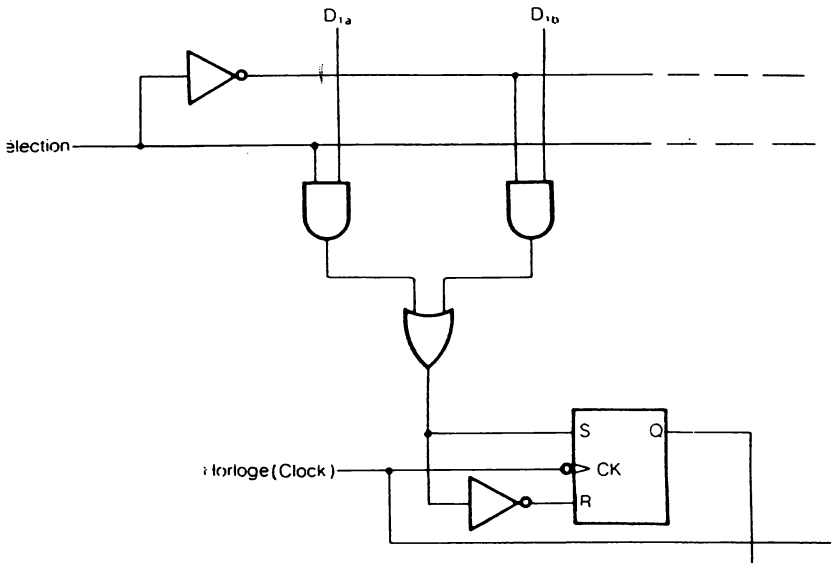


Fig. 11.15. — *Registre avec multiplexeur d'affichage associé. Schéma de principe d'un étage.*

Entrées		Sorties
Sélection	Horloge	Q_i^+
X	sans transition	Q_i
0	↑	D_{1b}
1	↑	D_{1a}

Fig. 11.16. — *Registre avec multiplexeur d'affichage associé. Table de vérité.*

4. LES MODES DE SORTIES

4.1. Sorties des bascules ou bistables

L'utilisateur dispose au niveau des broches du boîtier des sorties Q et $\bar{Q}$, ou des seules sorties Q. La plupart des registres intégrés sont du dernier type car cela réduit le nombre de broches nécessaires.

4.2. Sorties avec amplificateur séparateur

La figure 11.17 montre la cellule élémentaire d'un registre dont la sortie $\overline{Q}$ comporte un inverseur séparateur. Cet inverseur isole la sortie $\overline{Q}$ de l'extérieur. En effet, on constate quand on réalise des liaisons un peu longues en sorties de registre (passage d'une carte à une autre, point de test ramené sur une face avant de coffret) que les couplages entre fils peuvent provoquer un basculement intempestif d'une bascule par sa sortie. Si l'on place un circuit d'isolement, la bascule est protégée de tels couplages.

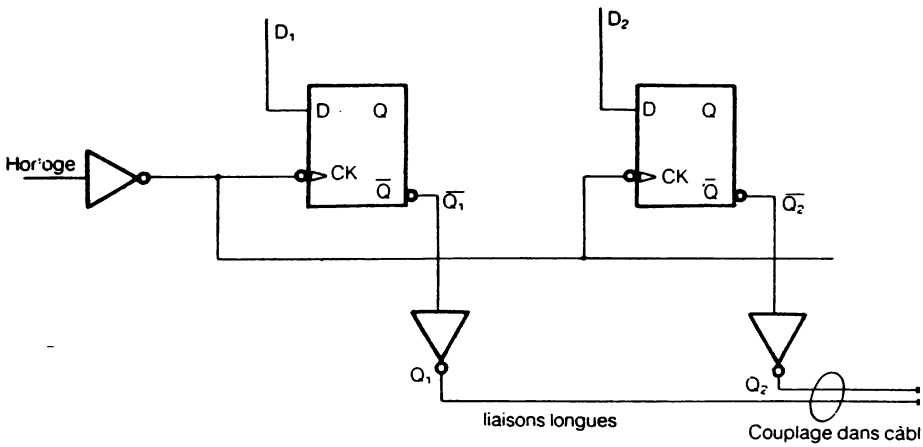


Fig. 11.17. — *Protection des sorties d'un registre contre les couplages.*

4.3. Sélection des sorties et interfaces trois états

L'utilisation de plus en plus fréquente d'architecture en bus dans les systèmes a conduit les constructeurs à proposer des registres à sorties trois états dont le schéma de principe est donné à la figure 11.18. Ce type de sortie trois états assure par principe la protection contre les basculements intempestifs par la sortie.

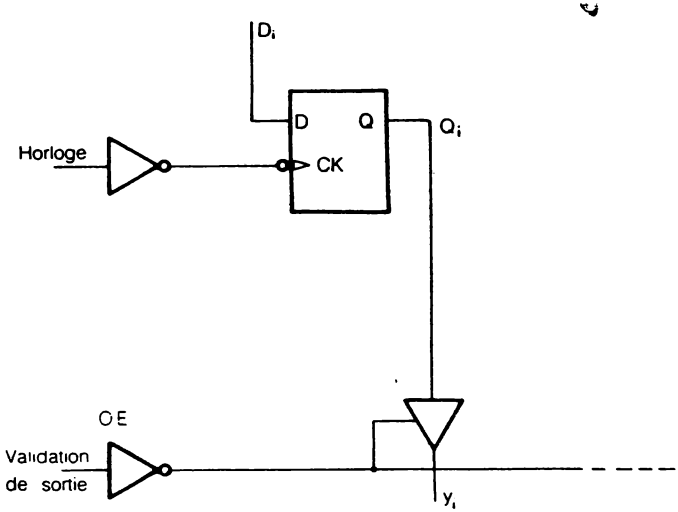


Fig. 11.18. — *Registre avec validation des sorties en mode trois états.*

5. EXEMPLE DE REGISTRE INTÉGRÉ : LE 74 LS 374

Ce registre (fig. 11.19) à huit étages présente un certain nombre des propriétés dégagées dans le paragraphe 1.

— Il dispose de commandes communes : horloge et validation des sorties.

— Il est de type à affichage sur front d'horloge.

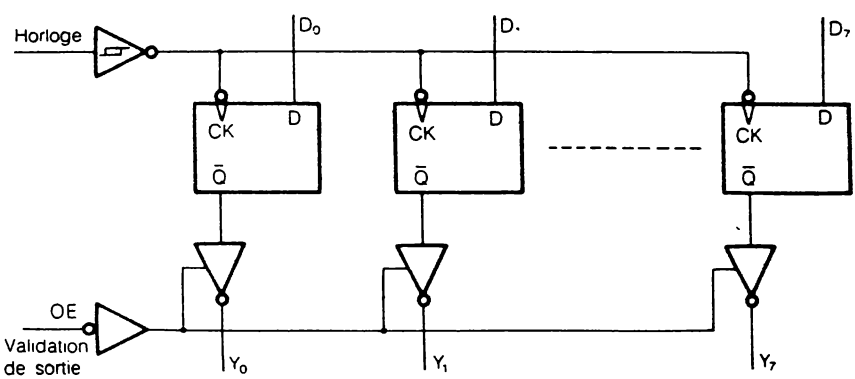


Fig. 11.19. — *Registre avec validation de sortie type 74 LS 374.*

— Il comporte une commande munie d'un trigger de Schmitt.

— Ces sorties sont de type trois états.

La figure 11.20 donne la table de vérité de ce registre.

Entrées			Etat interne	Sorties
OE	Horloge	D_i	$\overline{Q_i^+}$	Y_i
0	sans transition	X	$\overline{Q_i}$	Q_i
0	↑	D_i	$\overline{D_i}$	D_i
1	X	X	*	Z

Q_i : État mémorisé par le front montant de l'impulsion d'horloge precedente

X : État indifférent

* : État dépendant de l'horloge (voir lignes precédentes)

Z : État haute impedance

Fig. 11.20. — *Table de vérité du registre 74 LS 374.*

Afin de compléter la spécification de ce registre, nous donnons à la figure 11.21 le tableau des paramètres dynamiques et à la figure 11.22, celui des paramètres électriques.

Paramètres		MIN	NOM	MAX	Unité
Fréquence maximum d'horloge f max		30	50		MHz
Durée d horloge (CK)ou de validation (OE): t_w		15			ns
Temps de préaffichage des données : t_{su}		20 ↑			ns
Temps de maintien des données : t_h		0 ↑			ns
t_{PLH} depuis l'horloge			15	28	ns
t_{PHL} depuis l'horloge			19	28	ns
t_{PZH}	depuis l'entrée de validation OE		20	28	ns
t_{PZL}			21	28	ns
t_{PHZ}			12	20	ns
t_{PLZ}			14	25	ns

Fig. 11.21. — *Paramètres dynamiques du registre 74 LS 374.*

Paramètres		MIN	NOM	MAX	Unité
Tension d'alimentation V_{CC}		4.75	5	5.25	V
Courant de sortie, niveau haut I_{OH}				−2.6	mA
Courant de sortie niveau bas I_{OL}				20	µA
Température de fonctionnement T_A		0		70	°C
Tension d'entrée niveau haut V_{IH}		2			V
Tension d'entrée niveau bas V_{IL}				0,8	V
Tension de clamping V_{IK}				−1,5	V
Tension de sortie niveau haut V_{OH}		2,4	3,1		V
Tension de sortie niveau bas V_{OL}	I_{OL} = 4 mA		0,25	0,4	V
	I_{OL} = 8 mA		0,35	0,5	V
Courant d'entrée à la tension d'entrée maximum I_I				0,1	mA
Courant d'entrée, niveau haut I_{IH}				20	µA
Courant d'entrée, niveau bas I_{IL}				−0,4	mA
Courant de court-circuit I_{OS}		−30		−130	mA
Courant d'alimentation I_{CC}			27	40	mA

Fig. 11.22. — *Paramètres électriques du registre 74 LS 374.*

6. APPLICATIONS DES REGISTRES DE MÉMORISATION

Lorsque dans une application, il y a automodification ou modifications croisées simultanées, il faut impérativement utiliser des registres à bascules *D*. Quand ce n'est pas le cas, des bistables *D* ou des bascules *D* peuvent être choisies indifféremment. Dans certains cas, la propriété de transparence des registres à bistables *D* peut améliorer les performances d'un système.

6.1. Systèmes sans automodification

6.1.1. Registre d'interface entre deux sous-ensembles

Un registre peut servir de moyen de communication entre deux sous-ensembles (fig. 11.23). L'information calculée par le sous-ensemble *A* est mémorisée dans le registre *R* pour être ultérieurement utilisée par le sous-ensemble *B*. Un tel montage sous entend une synchronisation des actions

entre les sous-ensembles *A* et *B* de façon à exploiter le registre *R* quand son contenu est stabilisé.

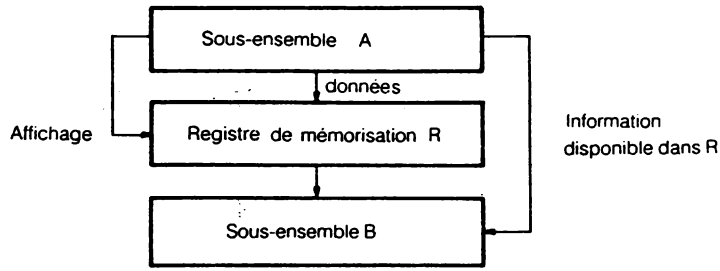


Fig. 11.23. — *Registre d'interface entre deux sous-ensembles.*

L'introduction d'un registre d'interface appelé encore *tampon* (ou buffer) entre deux sous ensembles permet d'améliorer les performances. Les figures 11.24 et 11.25 montrent une liaison entre deux sous-ensembles sans registre. Dans le premier cas, l'information apparaît en sortie de *A* et n'est maintenue qu'un certain temps. Cela impose au sous-ensemble *B* de prélever aussitôt cette information, ce qui peut être contraignant pour *B* (arrêt du travail en cours, mémorisation, reprise du travail interrompu). Dans le second cas, on n'impose plus à *B* de réagir immédiatement, et l'on maintient stable l'information sortante de *A* tant que *B* ne l'a pas prélevée. Ceci ralentit le travail de *A*. Enfin, la figure 11.26 montre qu'avec un registre d'interface, l'information est disponible pendant pratiquement tout le cycle et que le système *A* peut calculer la nouvelle information pendant que *B* enregistre et traite l'information présente dans le registre.

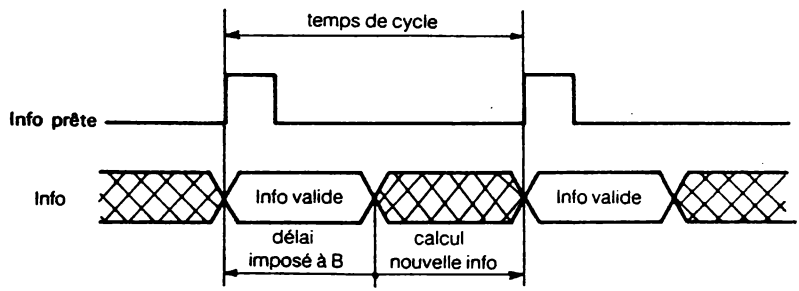


Fig. 11.24. — *Liaison sans registre tampon et temps imposé.*

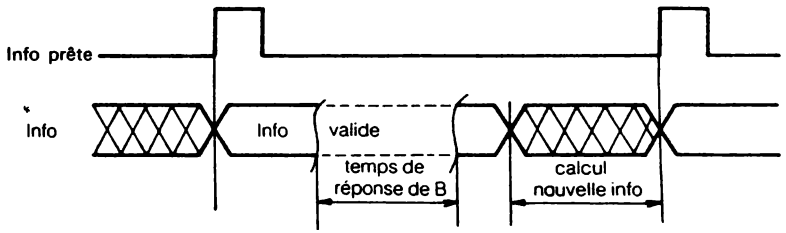


Fig. 11.25. — *Liaison sans registre tampon et attente.*

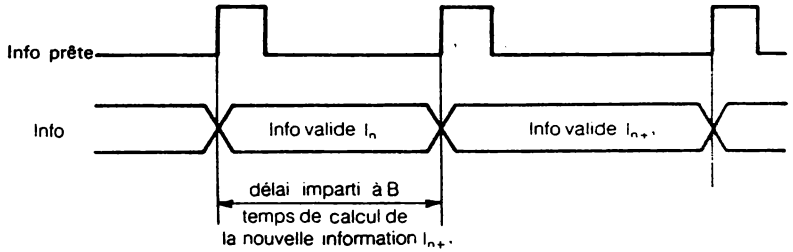


Fig. 11.26. — *Liaison avec registre tampon.*

6.1.2. Registre de stabilisation d'états instables

Supposons qu'à partir des données issues d'un processus, on exécute pour certaines combinaisons de ces données des traitements T_0, T_1, T_2, ... T_i dépendant des combinaisons 0, 1, 2, ... i de ces données. Le schéma naturel auquel on aboutit correspond à celui de la figure 11.27 : un décodeur dont les sorties valident les ensembles réalisant les traitements T_0, T_1, T_2, ... T_i. Or si les données et la logique effectuant ces traitements sont sans relation de phase précise, si la combinaison des données entrantes peut varier de façon quelconque par rapport aux horloges des logiques de traitement, on risque de commencer un traitement T_i pour continuer brusquement sur un traitement T_j. La figure 11.28 montre que l'on peut éviter cela en introduisant un registre de mémorisation qui stabilise les données et permet un calcul complet des actions pour une combinaison donnée. Il suffit qu'entre deux horloges h_0 de la figure 11.28, on ait le temps d'exécuter un traitement et que la fréquence de l'horloge h_0 soit supérieure à la fréquence de variation des données si l'on ne veut pas manquer une donnée entrante. Cet exemple illustre une règle générale : chaque fois que l'on prend des décisions à partir d'une information susceptible de varier, il convient de la stabiliser en la mémorisant.

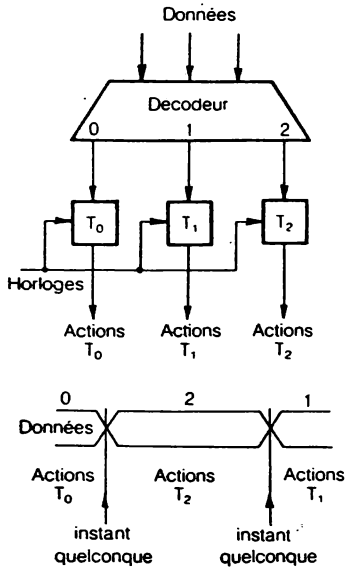


Fig. 11.27. — *Système non stabilisé.*

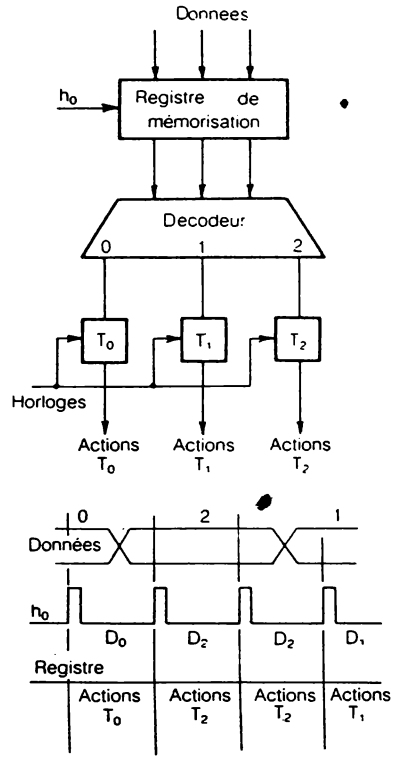


Fig. 11.28. — *Système stabilisé.*

6.2. Systèmes avec automodification des contenus de registres

6.2.1. Modèle de machine séquentielle simple

Une machine séquentielle simple peut se réaliser avec un registre, qui note les états présents de la machine, et deux blocs, calculant les sorties du système et l'état futur de la machine (fig. 11.29).

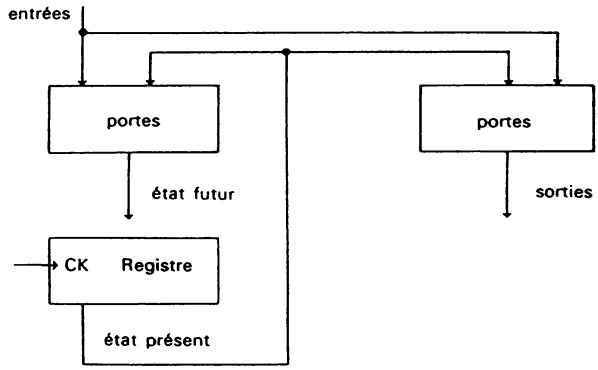


Fig. 11.29. — *Structure d'une machine séquentielle simple.*

6.2.2. Données et résultats d'un bloc de calcul arithmétique et logique

Les données et les résultats d'un bloc de calcul sont mémorisés par des registres comme le montre la figure 11.30.

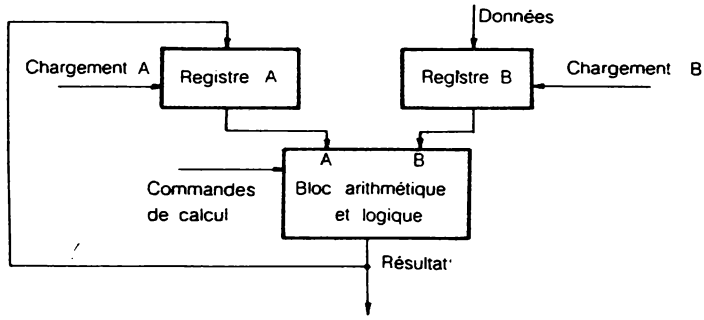


Fig. 11.31. — *Liaison inter-registres par bus.*

6.2.3. Liaison entre registres par un bus

Un ensemble complexe comprend de nombreux registres qui communiquent entre eux. Cette communication peut s'effectuer au moyen de liaison bus comme le montre la figure 11.31 avec des circuits trois états. Un décodeur ouvre les portes du registre source sélectionné sur le bus, tandis qu'un autre décodeur fournit une impulsion qui permet l'entrée du bus dans le registre destinataire choisi. Ce procédé, très économique en matériel, puisque certains registres possèdent en interne les portes trois états (voir

§ 4.3. et 5), ne permet cependant qu'un échange à la fois. Si plusieurs échanges simultanés étaient nécessaires, il faudrait plusieurs bus.

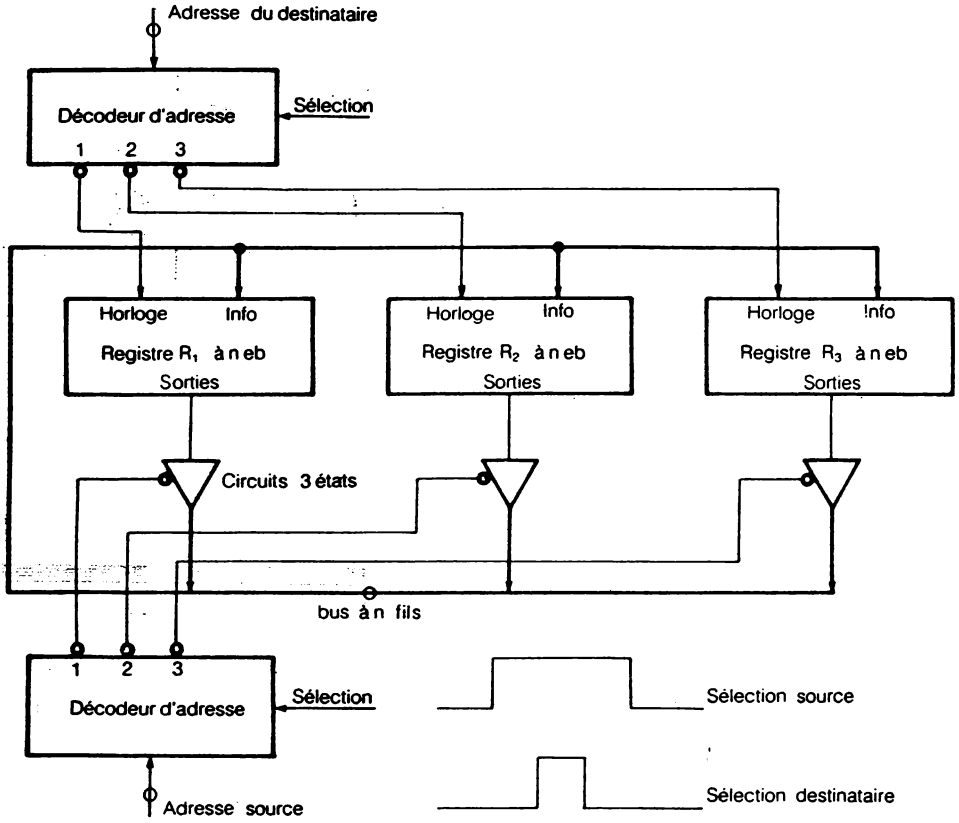


Fig. 11.31. — *Liaison inter-registres par bus.*

Annexe

REGISTRE DE MÉMORISATION PRINCIPAUX TYPES DE CIRCUITS MSI

1. TABLEAU SYNTHÉTIQUE

Lorsque l'on ouvre un catalogue de circuits intégrés, on est surpris par le nombre et la diversité des registres de mémorisation existants. Cependant, l'examen de leurs différentes caractéristiques telles que nous les avons dégagées au cours du chapitre 11 permet de les classer en quelques grandes familles. Nous avons ainsi classé 14 types de registres intégrés en quatre familles dans le tableau suivant (fig. 11.32).

Nous ne décrirons que quelques registres par classe. Le lecteur pourra en déduire facilement le fonctionnement des différents circuits intégrés du tableau et ceux non cités qu'il pourrait rencontrer dans les catalogues.

Type de boîtier	Nombre de broches du boîtier	Commandes communes						Type de mémorisation				MODE DE SORTIE				
		Horloge	Mise à zero clear	Validation sorties 3 états	Trigger sur commande	Validation des données affichées	Multiplexage des données affichées	Bistables (latches D)	Bascules D	RS synchrone	Nombre d'étages	Q	$\bar{Q}$	3 états	Ampli sur sorties	Sorties compatible MOS
74 LS 363	20	X		X	X			X			8	X		X		
74 LS 364	20	X		X	X				↑		8	X		X		
74 LS 373	20	X		X	X			X			8	X		X		X
74 LS 374	20	X		X	X				↑		8	X		X		X
74 LS 377	20	X				X			↑		8	X				
74 LS 378	16	X				X			↑		6	X				
74 LS 379	16	X				X			↑		4	X	X			
74 LS 273	20	X	X						↑		8	X				
74 LS 273 B	20	X	X						↑		8	X			X	
74 LS 174	16	X	X						↑		6	X				
74 LS 175	16	X	X						↑		4	X	X			
74 LS 298	16	X					X			↓	4	X				
74 LS 398	20	X					X			↑	4	X	X			
74 LS 399	16	X					X			↑	4	X				

Fig. 11.32. — *Principaux registres de mémorisation.*

2. REGISTRES A 8 ÉTAGES – SORTIES 3 ÉTATS ET TRIGGER DE SCHMITT SUR LA COMMANDE HORLOGE

(74 LS 363 – 74 LS 364 – 74 LS 373 – 74 LS 374)

2.1. Réalisé avec des bistables D (74 LS 363) et variante avec sorties compatibles MOS (74 LS 373)

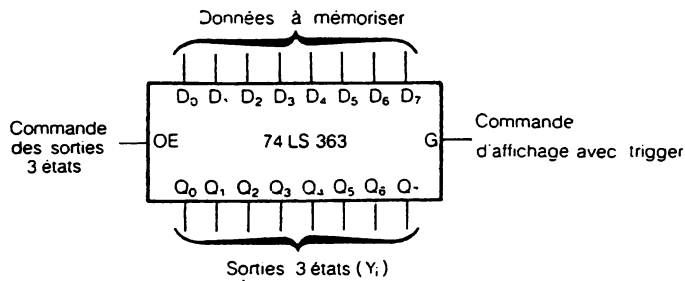


Fig. 11.33. — *Schéma de brochage du registre 74 LS 363.*

Entrees			Etat interne	Sorties
OE	G	D	$\bar{Q}_i$	Y_i
0	1	0	$\bar{D}$	D
0	0	X	$\bar{Q}_i$	$\bar{Q}_i$
1	X	X	*	Z

Q_i : État mémorisé par l'état $\bar{G}$ précédent
X : État indifférent
* : $\bar{Q}_i$ ou $\bar{D}$ suivant l'état de la commande G, voir lignes précédentes
Z : État haute impédance

Fig. 11.34. — *Table de vérité du registre 74 LS 363.*

2.2. Réalisé avec des bascules D (74 LS 364) et variante avec sorties compatibles MOS (74 LS 374)

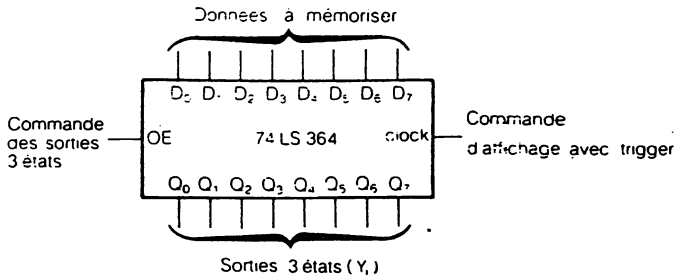


Fig. 11.35. — *Schéma de brochage du registre 74 LS 364.*

Entrées			Etat interne	Sorties
OE	Clock	D_i	$\bar{Q}_i$	Y_i
0	sans transition	X	$\bar{Q}_i$	Q_i
0	↑	D_i	$\bar{D}_i$	D_i
1	X	X	*	Z

Q_i : État mémorisé par le front montant de l'impulsion précédente
X : État indifférent
* : État dépendant de l'horloge (voir lignes précédentes)
Z : État haute impédance

Fig. 11.36. — *Table de vérité du registre 74 LS 364.*

3. REGISTRES AVEC VALIDATION DES DONNÉES AFFICHÉES (74 LS 377 – 74 LS 378 – 74 LS 379)

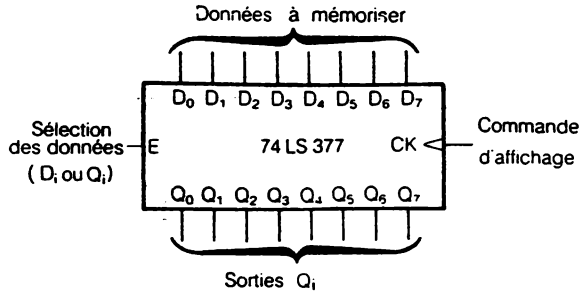


Fig. 11.37. – *Schéma de brochage du registre 74 LS 377.*

Entrées			Sorties
E	D_i	CK	Q_i^+
X	X	sans transition	Q_i
0	D_i	↑	D_i
1	X	↑	Q_i

X : Etat logique indifférent
↑ : Front montant de l'horloge
Q_i : Etat précédent de Q_i^+

Fig. 11.38. – *Table de vérité du registre 74 LS 377.*

4. REGISTRES AVEC REMISE A ZERO (CLEAR) (74 LS 273 – 74 LS 174 – 74 LS 175) ET VARIANTE AVEC AMPLIFICATEURS SÉPARATEURS SUR SORTIES Q (74 LS 273 B)

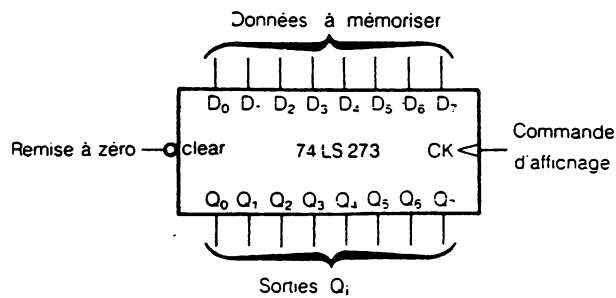


Fig. 11.39. – *Schéma de brochage du registre 74 LS 273.*

Entrées			Sorties
Clear	Clock	D_i	Q_i^*
0	X	X	0
1	sans transition	X	Q_i
1	↑	D_i	D_i

X : État logique indifférent
↑ : Front montant de l'horloge
Q_i État précédent de Q_i

Fig. 11.40. — *Table de vérité du registre 74 LS 273.*

5. REGISTRE AVEC MULTIPLEXAGE DES DONNÉES AFFICHÉES (74 LS 298 : AFFICHAGE SUR FONT DESCENDANT D'HORLOGE) (74 LS 398 -- 74 LS 399 : AFFICHAGE SUR FRONT MONTANT D'HORLOGE)

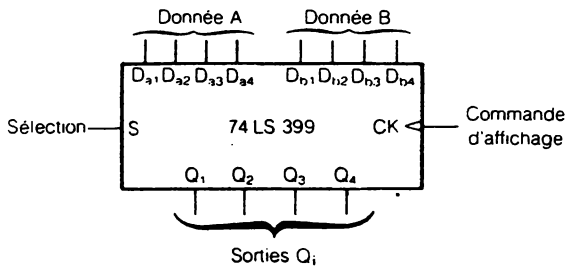


Fig. 11.41. — *Schéma de brochage du registre 74 LS 399.*

Entrées		Sorties
S	CK	Q_i^*
0	↑	a,
1	↑	b,
X	sans transition	Q_i

Q_i État précédent de Q_i
↑ : Front montant de l'horloge
X : État indifférent

Fig. 11.42. — *Table de vérité du registre 74 LS 399.*

Chapitre 12

LA FONCTION DE COMPTAGE

1. STRUCTURE DES COMPTEURS BINAIRES

Un compteur est un registre particulier (voir fig. 11.1) dont le contenu passe de la valeur n à $n + 1$ après application d'une impulsion de progression. Il existe deux types de structure: synchrone et asynchrone.

Dans la structure synchrone (fig. 12.1), l'impulsion de progression est appliquée directement à toutes les entrées horloges des bascules, tandis qu'un réseau combinatoire calcule les fonctions agissant sur les entrées synchrones (J, K ou D) associées. Dans cette structure toutes les bascules commutent en même temps, après un front déterminé du signal de progression.

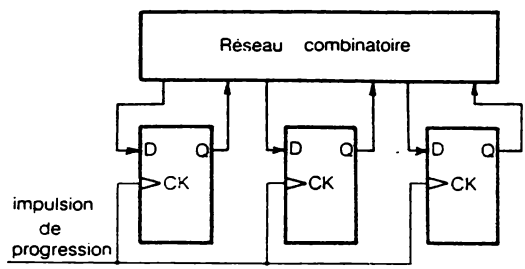


Fig. 12.1. — *Structure d'un compteur synchrone.*

Dans une structure asynchrone (fig. 12.2), on applique l'impulsion de progression sur l'entrée horloge du premier étage, les autres entrées d'horloge recevant des fonctions provenant de l'état des étages du compteur. De la sorte les étages ne basculent plus en synchronisme avec l'impulsion de progression mais après les transitions des étages qui les commandent. On dit que la progression est asynchrone.

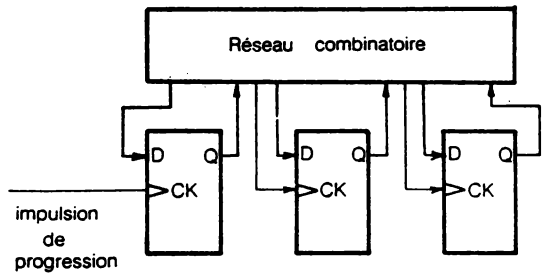


Fig. 12.2. — *Structure d'un compteur asynchrone.*

2. SYNTHÈSE DES COMPTEURS BINAIRES SANS CALCUL

La synthèse d'un compteur consiste à déterminer les fonctions de commande des bascules dans les structures des figures 12.1 et 12.2. On peut y parvenir sans calcul en analysant les transitions des étages. La figure 12.3 montre la succession des états pour un compteur à quatre étages Q_0, Q_1, Q_2, Q_3, affectés des poids 2^0, 2^1, 2^2, 2^3, et progressant en binaire.

2.1. Synthèse d'un compteur binaire synchrone

En analysant la figure 12.3, on constate que l'étage Q_n change d'état si tous les étages précédents sont à 1 : Q_0, Q_1 ... $Q_{n-1} = 1$ et que l'étage Q_n conserve son état dans le cas contraire. On appelle retenue de l'étage Q_n la fonction $r_n = Q_0\ Q_1 \ldots Q_{n-1}$. L'étage Q_0 qui n'a pas d'antécédent a une retenue $r_0 = 1$. Il change d'état systématiquement. Vérifions cette assertion pour l'étage Q_3 par exemple.

La condition de commutation $Q_2 = Q_1 = Q_0 = 1$ ne se rencontre qu'aux lignes 7 et 15 pour lesquelles Q_3 vaut respectivement 0 et 1. Aux lignes suivantes 8 et 0, Q_3 vaut respectivement 1 et 0. Q_3 a donc commuté. Pour

toutes les lignes autres que 7 et 15, la condition $Q_2 = Q_1 = Q_0 = 1$ n'est pas satisfaite et l'on constate que Q_3 conserve sa valeur. A la ligne 5 par exemple $Q_2 = 1$ $Q_1 = 0$, $Q_0 = 1$: Q_3 qui valait 0 à la ligne 5 vaut encore 0 à la ligne 6.

On peut écrire l'état futur de l'étage Q_n:

$$Q_n^+ = (Q_0\, Q_1 \cdots Q_{n-1}).\ \bar{Q}_n + (\overline{Q_0\, Q_1 \cdots Q_{n-1}}).\ Q_n$$

$$Q_n^+ = r_n\, \bar{Q}_n + \bar{r}_n\, Q_n = r_n \oplus Q_n.$$

Cette commutation s'effectue sous la dépendance de l'entrée horloge des bascules; une autre façon de noter le fonctionnement d'un compteur synchrone fait intervenir l'horloge:

$h \rightarrow r_n \oplus Q_n$ dans Q_n

Valeur	Q_3	Q_2	Q_1	Q_0	horloge
0	0	0	0	0	évolution du comptage sous la dépendance d'un front d'horloge
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	
0	0	0	0	0	

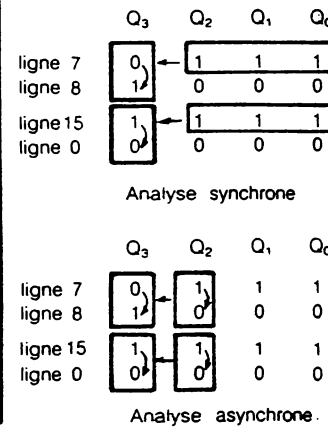


Fig. 12.3. — *Évolution des étages d'un compteur binaire.*

Selon la nature des bascules utilisées la commutation se produit sur un front montant ou descendant de l'horloge.

Si l'on réalise le compteur avec des bascules *JK*, on a, en rapprochant la formule du compteur et celle de la bascule:

$$Q_n^+ = r_n\, \bar{Q}_n + \bar{r}_n\, Q_n$$

$$Q_n^+ = J_n\, \bar{Q}_n + \bar{K}_n\, Q_n$$

d'où $J_n = K_n = r_n = Q_{n-1}\, Q_{n-2} \cdots Q_1\, Q_0$

La figure 12.4 montre le schéma correspondant sur quatre étages.

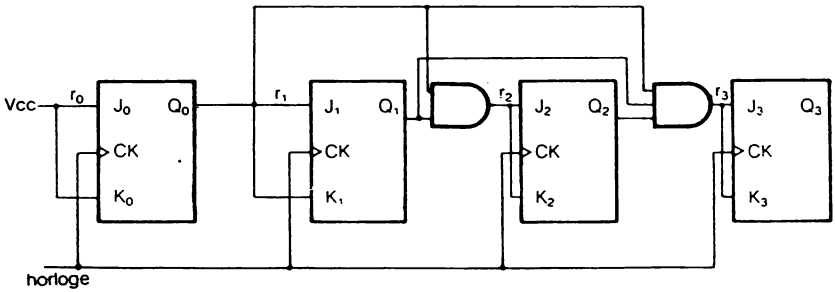


Fig. 12.4. — *Schéma de principe d'un compteur synchrone réalisé avec des bascules JK.*

Si l'on réalise le compteur avec des bascules D, on a :

$$Q_n^+ = r_n\ \bar{Q}_n + \bar{r}_n\ Q_n = r_n \oplus Q_n$$

$$Q_n^+ = D_n$$

d'où $D_n = r_n \oplus Q_n$

La figure 12.5 montre le schéma avec des bascules D.

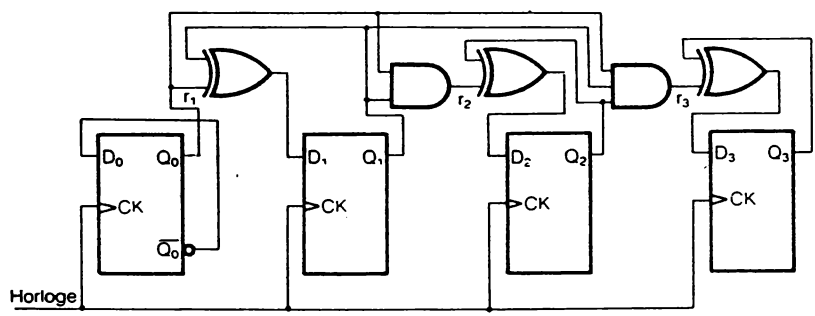


Fig. 12.5. — *Schéma de principe d'un compteur synchrone réalisé avec des bascules D.*

2.2. Synthèse d'un compteur binaire asynchrone

On constate en analysant les transitions de la figure 12.3 qu'un étage Q_n change d'état si l'étage précédent passe de 1 à 0 et conserve son état dans

les autres situations. L'étage Q_0, qui n'a pas d'antécédent, commute à chaque impulsion d'horloge. Vérifions cela pour l'étage Q_3.

Les seules transitions 1 vers 0 de Q_2 se trouvent aux lignes 7, 8 d'une part et 15, 0 d'autre part. On voit que la transition 1 vers 0 de Q_2 des lignes 7, 8 entraîne le passage de 0 vers 1 de Q_3 et que la transition 1 vers 0 de Q_2 des lignes 15, 0 entraîne le passage de 1 vers 0 de Q_3. Pour toutes les autres transitions de Q_2, on constate que Q_3 conserve son état.

On peut décrire le comportement des étages avec les équations :

$h \rightarrow \bar{Q}_0$ dans Q_0
$Q_0\downarrow \rightarrow \bar{Q}_1$ dans Q_1
$Q_1\downarrow \rightarrow \bar{Q}_2$ dans Q_2
$Q_2\downarrow \rightarrow \bar{Q}_3$ dans Q_3

Prenons les bascules JK, déclenchées sur front descendant. La première équation se réalise en mettant h sur l'entrée horloge et 1 sur J et K. Au front descendant de h l'étage Q_0 commute ($J_0 = K_0 = 1$). Les équations suivantes $Q_{n-1}\downarrow \rightarrow \bar{Q}_n$ dans Q_n se réalisent en mettant Q_{n-1} sur l'entrée horloge de Q_n et 1 sur J_n et K_n. Quand Q_{n-1} passe de 1 à 0 (front de descente), la bascule Q_n commute car $J_n = K_n = 1$. La figure 12.6 montre le schéma correspondant.

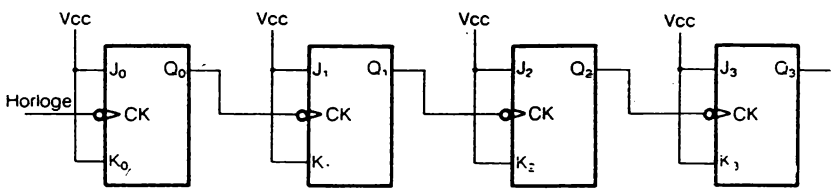


Fig. 12.6. — *Compteur asynchrone réalisé avec des bascules JK.*

Prenons maintenant des bascules D déclenchées sur front montant. La première équation se réalise en mettant h sur l'entrée horloge du premier étage et $\bar{Q}_0$ sur D_0. Quand h passe de 0 à 1 (front de montée) la bascule Q_0 commute ($D_0 = \bar{Q}_0$). Les équations suivantes $Q_{n-1}\downarrow \rightarrow \bar{Q}_n$ dans Q_n se réalisent en mettant $\bar{Q}_{n-1}$ sur l'entrée horloge de Q_n et $\bar{Q}_n$ sur l'entrée D_n. Quand Q_{n-1} pase de 1 à 0 (front de descente de Q_{n-1}), il y a un front montant sur $\bar{Q}_{n-1}$. A ce moment-là, Q_n commute ($D_n = \bar{Q}_n$).

La figure 12.7 donne le schéma correspondant.

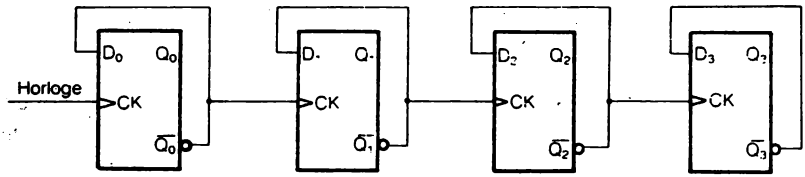


Fig. 12.7. — *Compteur asynchrone réalisé avec des bascules D.*

3. SYNTHÈSE D'UN COMPTEUR BINAIRE PAR LE CALCUL

Dans le paragraphe précédent nous avons déduit les schémas des compteurs d'une interprétation du tableau de comptage. On peut aussi calculer les compteurs. Pour cela on dresse un tableau de vérité donnant l'état présent, l'état futur de chaque étage et les états à appliquer sur les entrées des différents étages. Ces calculs ne s'appliquent qu'aux compteurs synchrones. La figure 12.8 donne le tableau de vérité d'un compteur à trois étages. Les états à appliquer sur J et K ont été déduits grâce à la table de transition. L'horloge ne figure pas dans le tableau. Il est implicite qu'une ligne donne une transition de l'état présent à l'état futur sous l'action d'une horloge.

État présent			État futur			Entrées					
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Table de transition

Fig. 12.8. — *Tableau de vérité d'un compteur binaire synchrone.*

A partir de ce tableau de vérité on dresse des tableaux de Karnaugh pour calculer les fonctions appliquées sur les entrées synchrones (fig. 12.9).

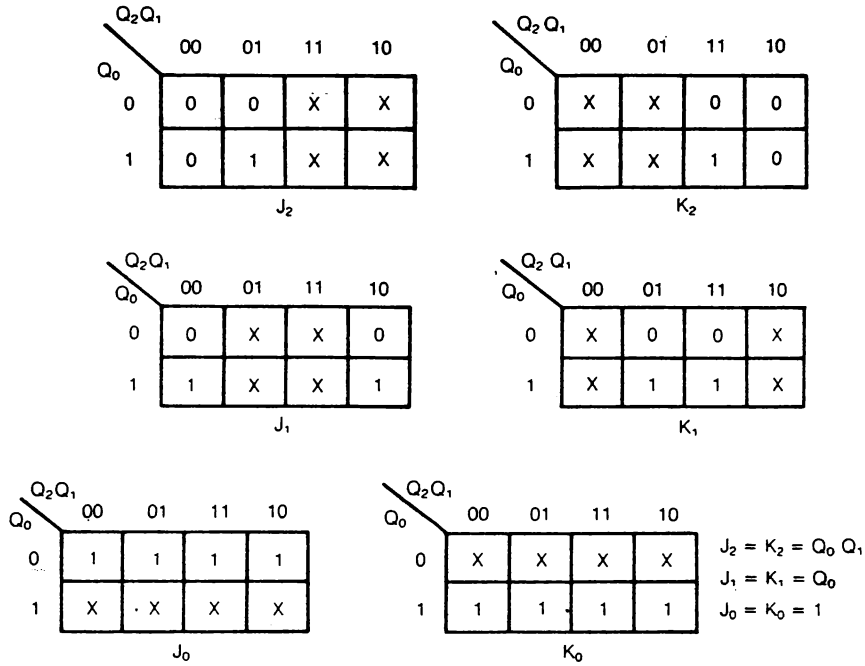


Fig. 12.9. — *Calcul des fonctions de commande.*

On retrouve les mêmes formules qu'au paragraphe 2, à partir de l'analyse du tableau de progression. Un calcul semblable pourrait être mené avec des bascules *D*. Cette méthode s'applique aussi au calcul d'un compteur à pas de progression quelconque.

4. LE COMPTAGE BINAIRE INCOMPLET

Jusqu'ici nous avons étudié des compteurs qui parcourent toutes les valeurs possibles de 0 à $2^n - 1$ si le compteur possède n étages, avec passage à 0 après la valeur $2^n - 1$. On a besoin également de compteurs à cycle incomplet, allant de 0 à N avec $N < 2^n - 1$ et passant à 0 après N. Sauf exception, on ne peut déduire de règle simple du tableau de progression. On peut cependant calculer le compteur selon la même méthode qu'au para-

graphe précédent ou comme nous le verrons plus loin utiliser des compteurs programmables qui évitent tout calcul. A titre d'exemple la figure 12.10 donne le tableau de vérité d'un compteur parcourant le cycle 0, 1, 2, 3, 4, 0, 1, 2 ... réalisé avec des bascules *D*. Son schéma est donné à la figure 12.11.

Etat présent			Etat futur			Entrées		
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	D_2	D_1	D_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0
1	0	1	X			X		
1	1	0		X			X	
1	1	1	X			X		

$D_2 = Q_1 Q_0$

$D_1 = Q_1 \overline{Q_0} + \overline{Q_1} Q_0 = Q_1 \oplus Q_0$

$D_0 = \overline{Q_2}\ \overline{Q_0}$

Q_0 \ Q_2Q_1	00	01	11	10
0	0	0	X	0
1	0	1	X	X

D_2

Q_0 \ Q_2Q_1	00	01	11	10
0	0	1	X	0
1	1	0	X	X

D_1

Q_0 \ Q_2Q_1	00	01	11	10
0	1	1	X	0
1	0	0	X	X

D_0

Fig. 12.10. — *Tableau de vérité d'un compteur à cycle incomplet.*

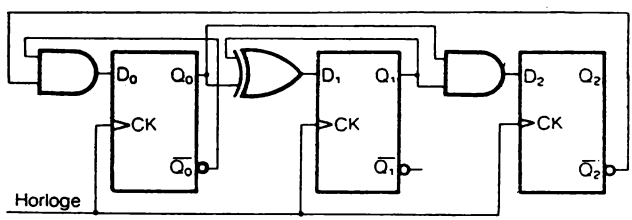


Fig. 12.11. — *Schéma d'un compteur à cycle incomplet.*

Les cycles incomplets les plus courants sont les cycles décimaux (comptage de 0 à 9) ou duodécimaux (comptage de 0 à 11).

5. LE DÉCOMPTAGE BINAIRE

On a besoin également dans les applications de décompteurs. Un décompteur est un registre dont le contenu passe de n à $n-1$ après application d'une impulsion. La figure 12.12 montre le tableau de décomptage pour quatre étages Q_0 Q_1 Q_2 Q_3. On constate, en s'en tenant à l'analyse synchrone qu'un étage Q_n commute si tous ceux de rang inférieur sont à 0 et conserve son état dans le cas contraire. Soit :

$$Q_n^+ = \bar{Q}_n \cdot \underbrace{(\overline{Q_{n-1}\, Q_{n-2} \cdots Q_1\, Q_0})}_{\text{commutation}} + Q_n \cdot \underbrace{(\overline{\bar{Q}_{n-1}\, \bar{Q}_{n-2} \cdots \bar{Q}_1\, \bar{Q}_0})}_{\text{maintien}}$$

Valeur	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
15	1	1	1	1
14	1	1	1	0
13	1	1	0	1
12	1	1	0	0
11	1	0	1	1
10	1	0	1	0
9	1	0	0	1
8	1	0	0	0
7	0	1	1	1
6	0	1	1	0
5	0	1	0	1
4	0	1	0	0
3	0	0	1	1
2	0	0	1	0
1	0	0	0	1
0	0	0	0	0

Fig. 12.12. — *Tableau de décomptage.*

La fonction $\bar{Q}_{n-1}\, \bar{Q}_{n-2} \cdots \bar{Q}_1\, \bar{Q}_0$ est appelée retenue comme dans le cas du compteur.

$$Q_n^+ = r_n\, \bar{Q}_n + \bar{r}_n\, Q_n = r_n \oplus \bar{Q}_n$$

Les formules générales du comptage et du décomptage sont identiques ; l'expression particulière de la retenue de décomptage s'obtient en faisant le produit des sorties inverses. Le schéma d'un décompteur synchrone s'obtient facilement à partir de celui du compteur synchrone de la figure 12.4 en remplaçant les connexions aux sorties Q par des connexions aux sorties $\bar{Q}$.

6. DIAGRAMME DES TEMPS D'UN COMPTEUR

6.1. Diagramme général

La figure 12.13 montre le diagramme général de trois registres: un compteur à trois étages progressant en binaire, un compteur à cycle incomplet réalisant le cycle 0, 1, 2, 3, 4, 0, 1, ... etc, étudié au paragraphe 4 et un décompteur binaire.

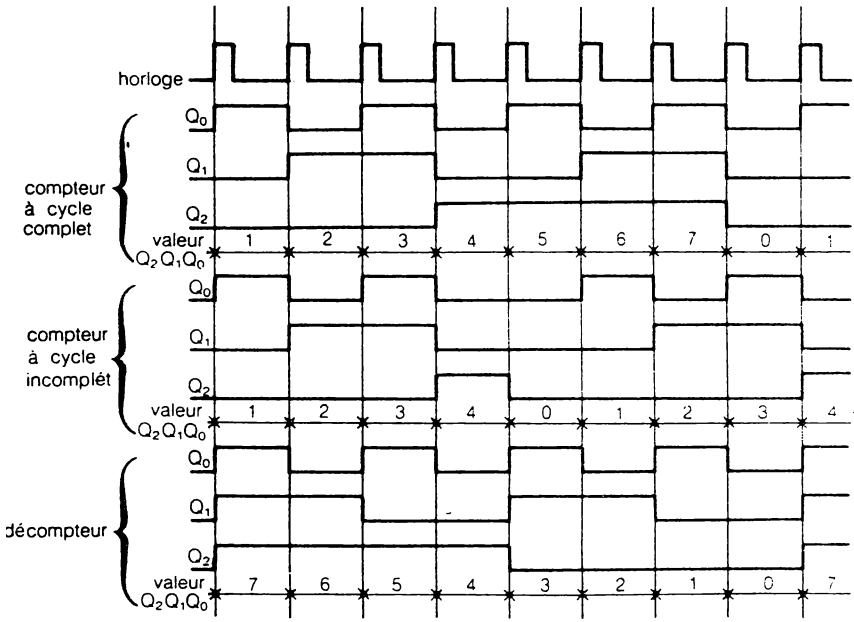


Fig. 12.13. — *Diagramme général d'un compteur, compteur à cycle incomplet, et décompteur.*

6.2. Les états transitoires

La figure 12.13 montre la succession des états des étages sans tenir compte des retards des circuits. La figure 12.14 traite du cas d'un compteur asynchrone à deux étages. Le premier étage commute sur le front montant de l'horloge et le second étage commute sur le front descendant du premier étage. La figure fait apparaître les temps de basculement t_{PLH} et t_{PHL}. On voit qu'en dehors des états stables il existe des états transitoires: au passage

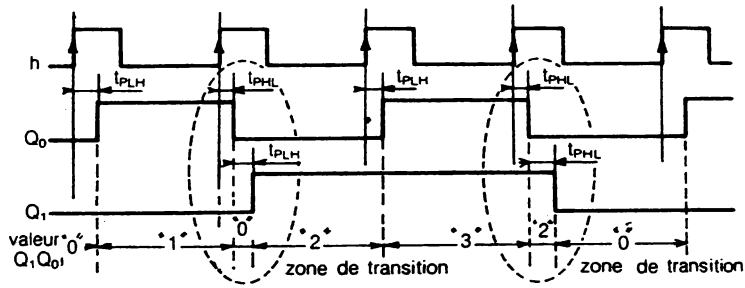


Fig. 12.14. — *Diagramme de progression d'un compteur asynchrone.*

de "1" à "2", le compteur passe par la position "0" car Q_0 est revenu à 0 avant que Q_1 ne soit passé à 1 ; au passage de "3" à "0", le compteur occupe provisoirement la position 2 car Q_0 est passé à 0 mais Q_1 ne l'a pas encore fait. Ces états transitoires durent le temps de basculement d'un étage.

La figure 12.15 montre le diagramme pour un compteur synchrone dont les deux étages commutent sur le front montant de l'horloge. On retrouve aussi des phénomènes transitoires sur les sorties d'un compteur synchrone au moment des transitions "1" vers "2" et "3" vers "0" mais on ne peut plus prévoir leur valeur, car cela dépend des durées respectives des retards t_{PHL}, t_{PLH} des étages dont on ne connaît que les valeurs maximums. La figure 12.15 montre deux cas. Si le t_{PHL} de Q_0 est supérieur au t_{PLH} de Q_1 : il apparaît la valeur transitoire "3" pendant le temps $t_{PHL} - t_{PLH}$. Par contre si le t_{PHL} de Q_0 est plus petit que le t_{PLH} de Q_1, il apparaît alors l'état transitoire "0" pendant le temps $t_{PLH} - t_{PHL}$. Pour un étage donné d'un compteur on ne connaît pas a priori la valeur t_{PHL} ou t_{PLH} (on ne connaît que la valeur maximum) de sorte que l'on est incapable de prévoir l'état transitoire pour un compteur synchrone : il a été noté x sur la figure 12.15. Les états transitoires d'un compteur synchrone durent $|t_{PHL} - t_{PLH}|$ donc moins longtemps que ceux d'un compteur asynchrone (t_{PLH} ou t_{PHL}). Leur durée peut cependant devenir comparable dans les cas suivants :

— étages appartenant à deux boîtiers, l'un étant plutôt lent et l'autre plutôt rapide. On a alors une différence $tp_{max} - tp_{min}$;

— dissymétries dans le circuit réalisant le décodage des états, certains étages étant retardés par des portes ;

— charges capacitives plus importantes sur certains étages ce qui dissymétrise les retards.

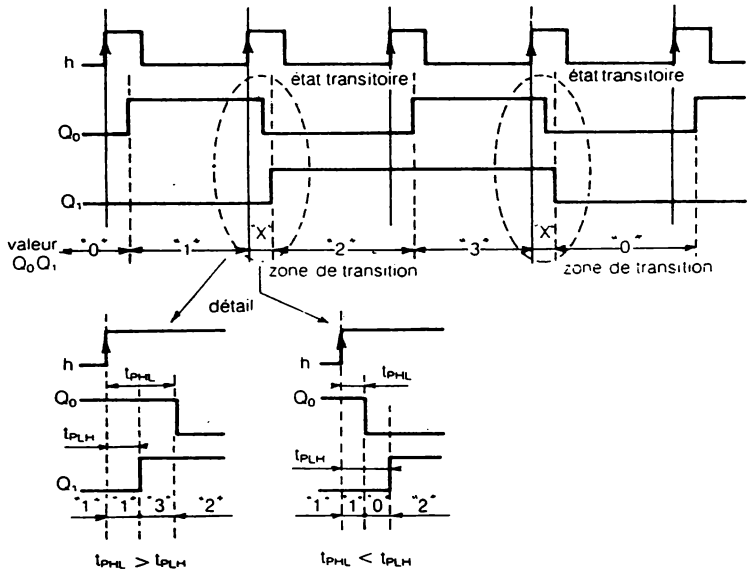


Fig. 12.15. — *Diagramme de progression d'un compteur synchrone et états transitoires.*

En conclusion que le compteur soit synchrone ou asynchrone il existe des états transitoires au moment de la commutation. Il ne faut pas exploiter les états du compteur pendant ces instants d'instabilités.

7. MISE EN CASCADE DES COMPTEURS

Un boîtier de compteur comprend généralement quatre étages. Pour constituer un compteur de plus grande taille, il faut associer plusieurs boîtiers en cascade.

7.1. Compteurs asynchrone

Pour un compteur asynchrone la liaison entre l'étage Q_{n-1} et Q_n s'écrit :

$Q_{n-1} \downarrow \rightarrow \overline{Q}_n$ dans Q_n

Si Q_{n-1} est le dernier étage d'un boîtier et Q_n le premier étage du boîtier suivant, il suffit de connecter Q_{n-1} sur l'entrée horloge de Q_n, celle-ci étant

sensible à un front de descente. La figure 12.16 montre la connexion pour trois boîtiers.

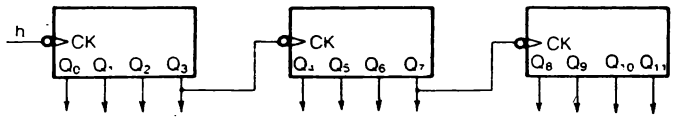


Fig. 12.16. — *Mise en cascade de trois compteurs asynchrones.*

7.2. Compteur synchrone

Examinons maintenant le cas d'un compteur synchrone à 16 étages: $Q_0\ Q_1 \ldots Q_{15}$, constitué avec des tranches de 4 eb. La tranche Q_0^3 compte systématiquement à chaque impulsion d'horloge. La tranche Q_4^7 compte seulement si $Q_0\ Q_1\ Q_2\ Q_3 = 1$ soit $Q_0^3 = 15$. Dans tous les autres cas elle garde sa valeur. La tranche Q_8^{11} compte seulement $Q_0\ Q_1\ Q_2\ Q_3\ Q_4\ Q_5\ Q_6\ Q_7 = 1$ soit $Q_0^3 = 15$ et $Q_4^7 = 15$. De même la tranche Q_{12}^{15} compte seulement si $Q_0^3 = Q_4^7 = Q_8^{11} = 15$. On a donc besoin d'un compteur un peu plus complexe que ceux étudiés dans les paragraphes précédents. Il doit être muni:

- d'une entrée V de validation de comptage. Si $V = 0$, le compteur ne compte pas; si $V = 1$, le compteur progresse;
- d'une sortie de retenue calculant la fonction $Q_0\ Q_1\ Q_2\ Q_3 = 15$. Ce compteur se déduit de celui étudié au paragraphe 2.1, dont l'équation était: $h \rightarrow r_n \oplus Q_n$ dans Q_n.

Il suffit, pour obtenir un compteur avec validation, de multiplier la retenue de chaque étage par V.

$$h \rightarrow V r_n \oplus Q_n \text{ dans } Q_n$$

Si $V = 1$, l'état du compteur dépend de la retenue comme au paragraphe 2.1. Si $V = 0$, $0 \oplus Q_n = Q_n$ de sorte que tous les étages sont inchangés.

La retenue sortante nécessite simplement une porte ET sur les quatre sorties. La figure 12.17 montre le schéma d'un tel compteur. La retenue sortante a été multipliée par V, ce qui facilite la connexion entre les boîtiers comme nous allons le voir.

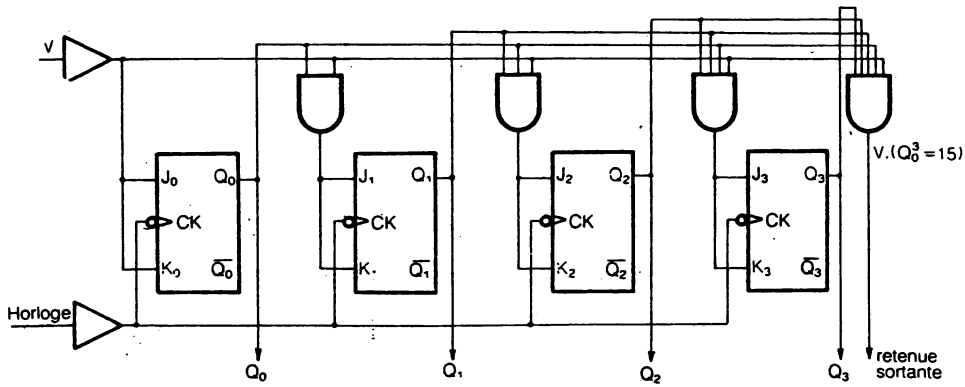


Fig. 12.17. — *Schéma d'un compteur avec validation de comptage et retenue sortante.*

La figure 12.18 montre le schéma de connexion entre les boîtiers. Il est particulièrement simple : même horloge pour tous les boîtiers et connexion de la retenue sortante à l'entrée V de deux boîtiers adjacents. Vérifions par exemple le comportement de Q_{12}, 1er étage du dernier boîtier. On vérifie facilement que la fonction appliquée sur l'entrée V du dernier boîtier vaut :

$$VAL.rs_0^3 \,.\, rs_4^7 \,.\, rs_8^{11} = VAL.(Q_0\, Q_1\, Q_2\, Q_3).(Q_4\, Q_5\, Q_6\, Q_7)\,. \\ (Q_8\, Q_9\, Q_{10}\, Q_{11}).$$

L'étage Q_{12} fonctionne donc comme prévu selon la formule :

$h\uparrow \rightarrow VAL.(Q_0\, Q_1 \,\ldots\, Q_{11}) \oplus Q_{12}$ dans Q_{12}.

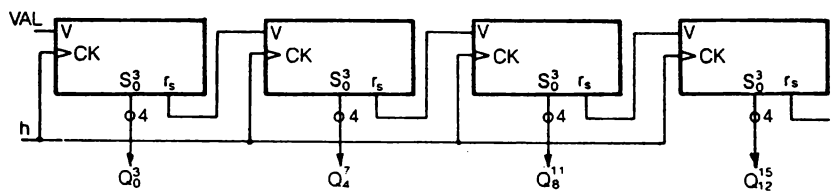


Fig. 12.18. — *Mise en cascade de compteurs synchrones.*

8. LE COMPTEUR PROGRAMMABLE

8.1. Principe

Jusqu'ici nous avons étudié un registre n'ayant que la seule fonction de comptage. On peut adjoindre d'autres fonctions qui étendent les possibilités du circuit. Ce sont par exemple :

- une remise à zéro qui met l'ensemble des bascules à 0 quel que soit l'état du compteur,
- une commande de programmation qui permet d'inscrire une information dans le registre quel que soit l'état du compteur,
- une validation du comptage (déjà vue pour la mise en cascade des compteurs synchrones), qui autorise ou non le comptage,
- une validation du comptage ou du décomptage.

Ces commandes auxiliaires agissent soit en liaison avec l'horloge générale du compteur soit indépendamment de celle-ci. Dans le premier cas il s'agit d'une commande synchrone et dans le second cas d'une commande asynchrone. Si le compteur dispose de plusieurs commandes, des règles de priorité existent entre elles. Passons en revue quelques types de compteur programmables.

8.2. Compteur asynchrone avec remise à zéro asynchrone

La figure 12.19 représente un compteur à progression asynchrone avec remise à zéro asynchrone. Le signal R agit sur les bascules indépendamment

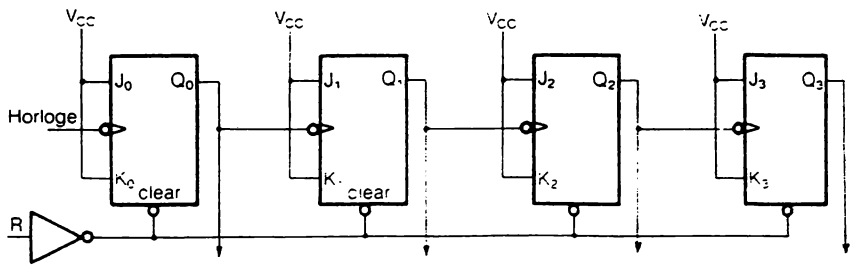


Fig. 12.19. — *Compteur asynchrone avec remise à zéro asynchrone.*

du signal de progression. Le fonctionnement de ce compteur se décrit par deux équations.

$h\downarrow \rightarrow +1$ dans Q_0^3

$R \rightarrow Q_0^3$ à 0

8.3. Compteur synchrone avec commandes synchrones

8.3.1. Définition

Soit un compteur pourvu des entrées suivantes:

- une horloge qui valide toutes les opérations affichées sur les autres entrées,
- une commande V validant la fonction comptage. Si $V = 1$ le compteur progresse, si $V = 0$ le compteur conserve son état,
- une commande P de programmation à laquelle sont associées des entrées d'informations I_0, I_1, ... I_n. Quand $P = 1$ l'information $I_0\ I_1\ \ldots\ I_n$ est inscrite dans les bascules $Q_0\ Q_1\ \ldots\ Q_n$ du compteur,
- une commande R de remise à 0. Quand $R = 1$, les bascules passent à 0.

Il existe une priorité entre les commandes: R l'emporte sur P qui l'emporte sur V.

Pour étudier un tel compteur il faut en principe établir un tableau de vérité comprenant:

- en entrée les états présents $Q_0\ Q_1 \ldots Q_n$, les commandes R, P, V, les informations $I_0\ I_1\ \ldots\ I_n$,
- en sortie les états futurs et les fonctions à appliquer sur les entrées D_0, D_1 ... D_n ou $J_0\ K_0, J_1\ K_1$, etc. Même pour un compteur à 4 étages le nombre de lignes est trop grand pour une mise en œuvre pratique. Si l'on ne s'intéresse qu'à un seul étage Q_i l'étude par un tableau de vérité reste cependant possible. Pour un étage Q_i interviennent les commandes R, P, V, l'information I_i et la retenue r_i de l'étage Q_i (r_i étant égal à $Q_1\ Q_0 \ldots Q_{i-2}\ Q_{i-1}$).

Avec les règles de priorité données plus haut on établit facilement le tableau de vérité de l'étage Q_i (fig. 12.20), l'horloge étant sous-entendue:

- si la remise à 0 est active ($R = 1$), l'étage Q_i passe à 0: $Q_i^+ = 0$,

- si la remise à 0 est inactive ($R = 0$) et si la programmation est active ($P = 1$) la valeur I_i passe dans Q_i: $Q_i^+ = I_i$,
- si la remise à 0, la programmation et la validation de comptage sont inactives ($R = P = V = 0$), l'étage conserve son état: $Q_i^+ = Q_i$,
- si la remise à 0, la programmation sont inactives et la validation active mais la retenue à 0 ($R = P = 0$, $V = 1$, $r_i = 0$), l'étage conserve aussi son état: $Q_i^+ = Q_i$,
- si la remise à 0, la programmation sont inactives, la validation de comptage active et la retenue à 1 ($R = P = 0$, $V = r_i = 1$), l'étage commute: $Q_i^+ = \overline{Q}_i$.

8.3.2. Table de vérité et équations logiques

A partir du tableau on obtient:

$$Q_i^+ = \overline{R}\,P\,I_i + \overline{R}\,\overline{P}\,\overline{V}\,Q_i + \overline{R}\,\overline{P}\,V\,\overline{r}_i\,Q_i + \overline{R}\,\overline{P}\,V\,r_i\,\overline{Q}_i$$

$$Q_i^+ = \overline{R}\,P\,I_i + \overline{R}\,\overline{P}\,[Q_i(V\,\overline{r}_i + \overline{V}) + V r_i\,\overline{Q}_i]$$

$$Q_i^+ = \overline{R}\,P\,I_i + \overline{R}\,\overline{P}\,[Q_i(\overline{r}_i + \overline{V}) + V\,r_i\,\overline{Q}_i]$$

$$Q_i^+ = \overline{R}\,P\,I_i + \overline{R}\,\overline{P}\,[Q_i\,\overline{V\,r_i} + \overline{Q}_i\,V\,r_i]$$

$$Q_i^+ = \overline{R}\,P\,I_i + \overline{R}\,\overline{P}\,(Q_i \oplus V\,r_i)$$

On peut noter que si $R = 0$, $P = 0$ et $V = 1$, on se retrouve dans le cas d'un compteur synchrone simple: $Q_i^+ = r_i \oplus Q_i$.

R	P	I_i	V	r_i	Q_i^+	commentaires
1	X	X	X	X	0	R prioritaire.
0	1	I_i	X	X	I_i	P prioritaire.
0	0	X	0	X	Q_i	V prioritaire comptage inhibé
0	0	X	1	0	Q_i	V prioritaire: comptage autorisé mais pas de retenue
0	0	X	1	1	$\overline{Q}_i$	V prioritaire: comptage autorisé avec retenue → Commutation de l'étage Q_i

Fig. 12.20. — *Tableau de vérité de l'étage Q_i d'un compteur programmable.*

La figure 12.21 donne le schéma d'un compteur programmable établi d'après la formule $Q_i^+ = \overline{R}PI_i + \overline{R}\,\overline{P}\,(Q_i \oplus Vr_i)$. La commande de l'étage Q_1 y est détaillée. Le fonctionnement de ce compteur se décrit aussi:

— soit par des équations ou interviennent l'horloge,

$R.\ h\uparrow \rightarrow Q_0^3$ à 0 (1).

$\bar{R}.P.h\uparrow \rightarrow I_0^3$ dans Q_0^3 (2)

$\bar{R}.\bar{P}.V.h\uparrow \rightarrow +\ 1$ dans Q_0^3 (3)

$V.Q_0.Q_1.Q_2.Q_3 =$ retenue.

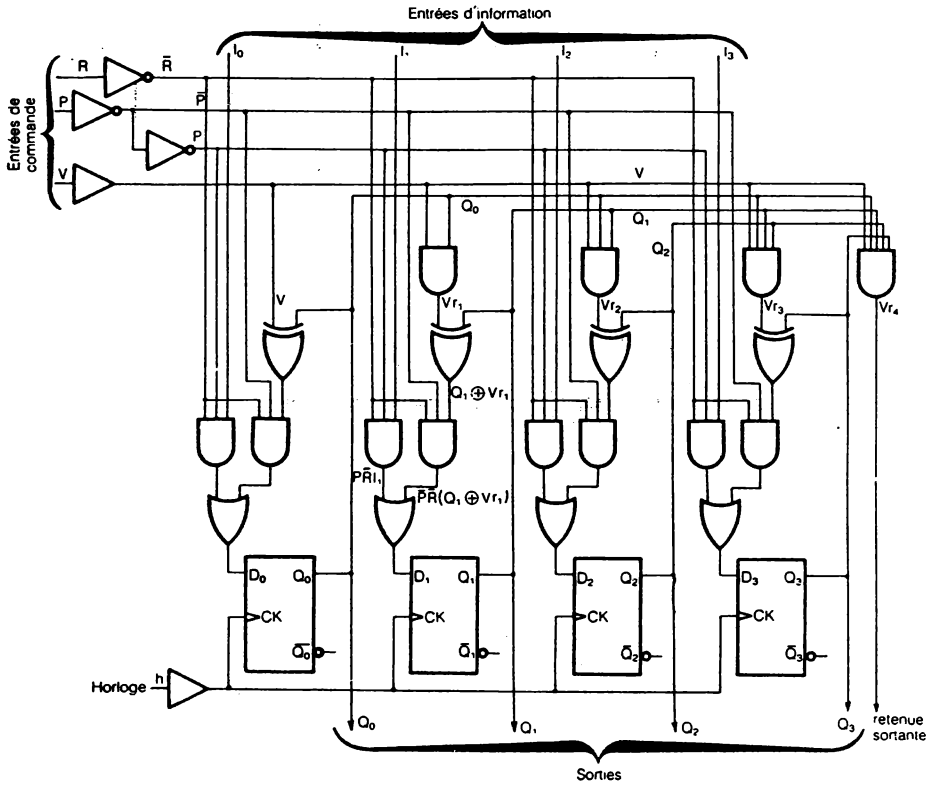


Fig. 12.21. — *Schéma d'un compteur programmable entièrement synchrone.*

— soit par un tableau de vérité (fig. 12.22) où apparaît l'horloge.

R	P	V	I_0^3	h	Q_0^{3-}
X	X	X	X	0 ou ↑	Q_0^3
0	0	0	X	X	Q_0^3
1	X	X	X	↑	0
0	1	X	a_0^3	↑	a_0^3
0	0	1	X	↑	Q_0^3+1

Fig. 12.22. — *Tableau de vérité d'un compteur programmable entièrement synchrone.*

8.4. Compteur synchrone avec commandes synchrones et asynchrones

Un compteur à progression synchrone peut avoir certaines commandes synchrones et d'autres asynchrones. La figure 12.23 montre un compteur ayant les mêmes commandes que celui du paragraphe précédent mais dans lequel P, V sont synchrones et R asynchrone. Les équations de description diffèrent légèrement.

$R \rightarrow Q_0^3$ à 0 (mise à zéro prioritaire asynchrone)

$P.h\uparrow \rightarrow I_0^3$ dans Q_0^3

$\overline{P}.V.h\uparrow \rightarrow +\ 1$ dans Q_0^3

$V.Q_0.Q_1.Q_2.Q_3 =$ retenue.

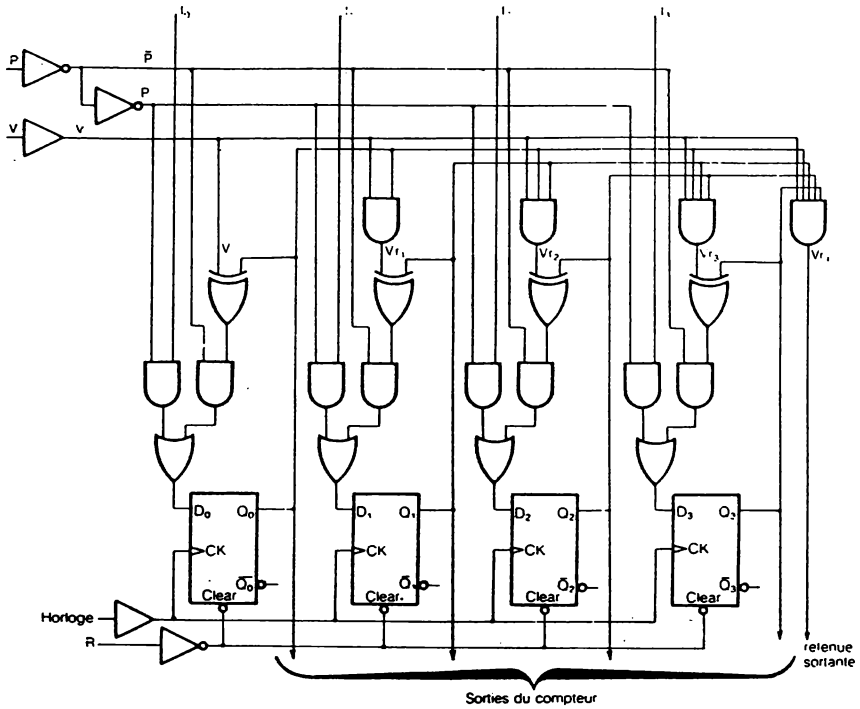


Fig. 12.23. — *Schéma d'un compteur programmable avec remise à zéro asynchrone.*

Le tableau de vérité de la figure 12.24 décrit également ce compteur.

R	P	V	I_0^3	h	Q_0^{3+}
0	X	X	X	0 ou 1	Q_0^3
0	0	0	X	X	Q_0^3
1	X	X	X	X	0
0	1	X	a_0^3	↑	a_0^3
0	0	1	X	↑	$Q_0^3 + 1$

Fig. 12.24. — *Tableau de vérité d'un compteur programmable avec progression et programmation synchrones et remise à zéro asynchrone.*

D'autres combinaisons de commandes sont possibles par exemple une programmation asynchrone. Le lecteur les trouvera dans l'annexe de ce chapitre.

9. PROPRIÉTÉS GÉNÉRALES DES COMPTEUR INTÉGRÉS

Les paragraphes précédents ont défini les propriétés des compteurs à cycle complet et incomplet, des décompteurs, des compteurs programmables. Aujourd'hui, sauf exception, il n'y a plus à concevoir de compteurs. Ils existent sous forme intégrée et il faut savoir les utiliser. Nous dégageons dans ce qui suit les propriétés générales des compteurs intégrés.

9.1. Comptage synchrone et asynchrone. Décomptage

Il existe des compteurs intégrés synchrones et asynchrones. La fonction de décomptage seule n'existe pas et se trouve regroupée avec la fonction comptage dans un circuit unique.

9.2. Structure et taille

Les compteurs utilisent des bascules *JK*, *D*, *RS*. La taille est généralement de 4 eb et le comptage peut être binaire (cycle complet 0 à 15 sur 4 eb), décimal (cycle incomplet de 0 à 9 sur 4 eb), ou duodécimal (cycle incomplet de 0 à 11 sur 4 eb).

9.3. Les commandes d'un compteur

On trouve la mise à 0, la mise à 9 (pour les comptages décimaux) la programmation à une valeur quelconque, le choix du sens de comptage (comptage ou décomptage). Le choix du sens de comptage se réalise soit par deux horloges indépendantes (une pour le comptage, une pour le décomptage) soit par une seule horloge et une commande précisant le sens. Des règles de priorité existent entre ces commandes qui peuvent être synchrones ou asynchrones. On trouve :

— des compteurs à progression asynchrone et commande asynchrone,

— des compteurs à progression synchrone et commande asynchrone,

— des compteurs à progression synchrone et commande synchrone.

9.4. Les sorties

On dispose le plus souvent des sorties directes Q, parfois de Q et $\overline{Q}$. Il peut exister des amplificateurs de sortie, ce qui isole l'étage du reste du montage et assure une protection contre les basculements par la sortie et éventuellement une sortance plus élevée. Les compteurs synchrones disposent d'une sortie retenue pour la mise en cascade.

10. EXEMPLES DE COMPTEURS INTÉGRÉS

Le lecteur trouvera en annexe de ce chapitre une synthèse des compteurs existants dans la famille 74 LS classée par caractéristiques principales. Nous étudions dans le détail seulement trois types de compteurs, mais l'ensemble des informations fournies dans cette description et dans les annexes permet aux utilisateurs d'analyser eux-même le fonctionnement d'un compteur intégré quelconque.

10.1. Le compteur binaire asynchrone 74 LS 93

C'est un compteur très simple. La figure 12.25 donne son schéma bloc. Il comprend :

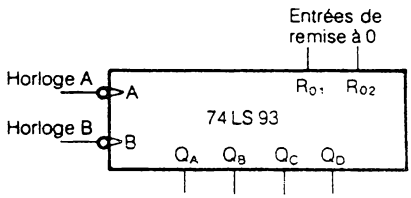


Fig. 12.25. — *Compteur intégré 74 LS 93.*

- quatre étages Q_A, Q_B, Q_C, Q_D répartis en deux sous-ensembles Q_A d'une part et Q_B, Q_C, Q_D d'autre part,
- une entrée d'horloge A qui sert à incrémenter l'étage Q_A,
- une entrée d'horloge B qui sert à incrémenter les trois étages Q_B, Q_C, Q_D,
- deux entrées de mise à 0, R_{01} et R_{02}. Quand ces deux entrées sont simul-

tanément à 1, les quatre étages Q_A, Q_B, Q_C, Q_D passent à 0. Ces commandes de remise à 0 sont prioritaires par rapport aux horloges A et B.

Le fonctionnement de ce compteur se décrit par les équations :

$A\downarrow\rightarrow+$ 1 dans Q_A (ou $\overline{Q}_A$ dans Q_A)

$B\downarrow\rightarrow+$ 1 dans Q_B^D

$R_{01}.\ R_{02}\rightarrow Q_A^D$ à 0.

On constitue un compteur à quatre étages en réunissant la sortie Q_A à l'entrée horloge B (voir paragraphe 7 sur la mise en cascade).

Il existe une variante 74 LS 90 et duodécimale 74 LS 92 de ce compteur.

10.2. Le compteur binaire synchrone 74 LS 163A

10.2.1. Présentation

C'est un compteur programmable complexe (fig. 12.26). Il comprend :

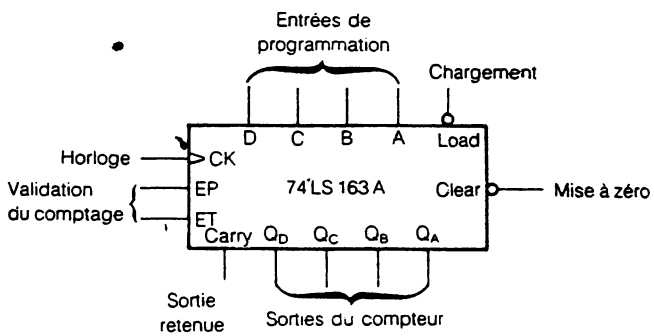


Fig. 12.26. – *Compteur intégré 74 LS 163 A.*

- quatre étages Q_A, Q_B, Q_C, Q_D,
- une entrée horloge CK,
- une entrée de mise à zéro clear, pour les quatre étages Q_A, Q_B, Q_C, Q_D,
- une entrée de chargement load pour programmer (afficher, charger) les quatre étages Q_A, Q_B, Q_C, Q_D avec la valeur présente aux quatre entrées de programmation A, B, C, D,

— deux entrées de validation de comptage *EP* et *ET*, qui autorisent le comptage si elles sont toutes deux à 1,

— une sortie de retenue qui est en 1 si *ET* vaut 1 et si les quatre étages Q_A, Q_B, Q_C, Q_D sont à 1 (valeur 15 pour le compteur). Cette sortie sert à valider les entrées de comptage du boîtier suivant si la taille du compteur dépasse 4 eb.

Ce compteur est entièrement synchrone :

— toutes les commandes dépendent de l'horloge (comptage, mise à zéro et chargement). La commutation s'effectue au front montant de l'horloge,

— l'entrée clear (active en inverse) est prioritaire par rapport aux entrées load et *EP*, *ET*,

— l'entrée load (active en inverse) est prioritaire par rapport aux entrées *EP* et *ET*.

On peut décrire le fonctionnement de ce compteur par le tableau de vérité de la figure 12.27 ou par les équations suivantes :

mise à zéro : $h\uparrow.\ \overline{\text{clear}} \rightarrow 0$ dans Q_A^D

chargement : $h\uparrow.\ \text{clear}.\ \overline{\text{load}} \rightarrow A$ à D dans Q_A^D

comptage : $h\uparrow.\ \text{clear}.\ \text{load}.\ EP.ET \rightarrow +1$ dans Q_A^D

inhibition : $h\uparrow.\ \text{clear}.\ \text{load}.\ \overline{EP.ET} \rightarrow Q_i$ dans Q_i (recopie)

retenue : $ET.(Q_A^D = 15) = \text{carry}$;

Horloge	Clear	Load	EP.ET	A B C D	$(Q_A\ Q_B\ Q_C\ Q_D)^+$	Résultat des commandes
niveau 0 ou 1	X	X	X	X	$(Q_A\ Q_B\ Q_C\ Q_D)$	Pas de front d'horloge : pas d'actions
↑	0	X	X	X	0 0 0 0	Mise à 0 (clear = 0)
↑	1	0	X	A B C D	A B C D	Affichage (Load = 0)
↑	1	1	0	X	$(Q_A\ Q_B\ Q_C\ Q_D)$	Inhibition comptage (EP, ET = 0,)
↑	1	1	1	X	$(Q_A\ Q_B\ Q_C\ Q_D) + 1$	Comptage

Fig. 12.27. — *Tableau de vérité du compteur 74 LS 163 A.*

Il existe des variantes de ce compteur : compteurs décimaux et mise à 0 asynchrone au lieu de synchrone (voir annexe). Ainsi le compteur décimal 74 LS 160 A, qui a une mise à zéro asynchrone se décrit par :

clear → 0 dans Q_A^D (mise à zéro prioritaire asynchrone)

$h\uparrow$. $\overline{\text{load}}$ → A à D dans Q_A^D

$h\uparrow$. load. $EP.ET$ → + 1 dans Q_A^D

$h\uparrow$. load. $\overline{EP.ET}$ → Q_i dans Q_i (recopie)

ET. Q_D. Q_A = Carry.

La sortie retenue vaut 1 quand le compteur atteint 9 ($Q_A\ Q_D = 1$) et si $ET = 1$.

10.2.2. Paramètres électriques et dynamiques du compteur 74 LS 163A

Les figures 12.28 et 12.29 donnent les caractéristiques électriques et dynamiques du compteur 74 LS 163 A relevées dans un catalogue. Il ne s'agit que d'un extrait car le catalogue précise en outre les conditions de mesure de tous les paramètres. Pour les paramètres électriques il faut noter que les entrées load, clock, ET et clear représentent deux charges de la technologie *LS*.

Pour les paramètres dynamiques on notera la variété des temps de propagation puisque les sorties peuvent réagir à plusieurs entrées.

Paramètres		MIN	NOM	MAX	Unité
Tension d'alimentation V_{CC}		4,75	5	5,25	V
Courant de sortie, niveau haut I_{OH}				-400	µA
Courant de sortie niveau bas I_{OL}				8	mA
Température de fonctionnement T_A		0		70	°C
Tension d'entrée niveau haut V_{IH}		2			V
Tension d'entrée niveau bas V_{IL}				0,8	V
Tension d'écrêtage V_{IK}				-1,5	V
Tension de sortie niveau haut V_{OH}		2,7	3,4		V
Tension de sortie niveau bas V_{OL}			0,35	0,5	V
Courant d'entrée I_I à la tension maximum d'entrée	A, B, C, D, EP			0,1	mA
	Load, Clock ET, Clear			0,2	mA
Courant d'entrée, niveau haut I_{IH}	A, B, C, D, EP			20	µA
	Load, Clock ET, Clear			40	µA
Courant d'entrée, niveau bas I_{IL}	A, B, C, D, EP			-0,4	mA
	Load, Clock ET, Clear			-0,8	mA
Courant de court-circuit I_{OS} en sortie		-20		-100	mA
Courant d'alimentation I_{CCH} toutes les entrees au niveau haut			18	31	mA
Courant d'alimentation I_{CCL} toutes les entrées au niveau bas			19	32	mA

Fig. 12.28. — *Paramètres électriques du compteur 74 LS 163 A.*

Paramètres			MIN	NOM	MAX	Unité
Fréquence de l'horloge f_{Clock}			0		25	MHz
Durée de l'horloge $t_{w\ (Clock)}$			25			nS
Durée de la remise à zéro $t_{w\ (Clear)}$			25			nS
Temps de préaffichage t_{su}		A, B, C, D EP, ET Load, Clear	20			nS
Temps de maintien t_h		A, B, C, D EP, ET Load, Clear	0			nS
retards	depuis	vers				
t_{PLH}	horloge	retenue		20	35	nS
t_{PHL}				18	35	nS
t_{PLH}	horloge (Load au niveau haut)	n'importe quelle sortie Q		13	24	nS
t_{PHL}				18	27	nS
t_{PLH}	horloge (Load au niveau bas)	n'importe quelle sortie Q		13	24	nS
t_{PHL}				18	27	nS
t_{PLH}	Enable T	retenue		9	14	nS
t_{PHL}				9	14	nS
t_{PHL}	Clear	n'importe quelle sortie Q		20	28	nS

Fig. 12.29. — *Paramètres dynamiques du compteur 74 LS 163 A.*

Pour les paramètres dynamiques on notera la variété des temps de propagation puisque les sorties peuvent réagir à des entrées.

10.3. Le compteur-décompteur 74LS169A

La figure 12.30 montre le schéma de principe du compteur-décompteur 74 LS 169 A. Comme pour le 74 LS 163, quelques équations décrivent le comportement de ce compteur.

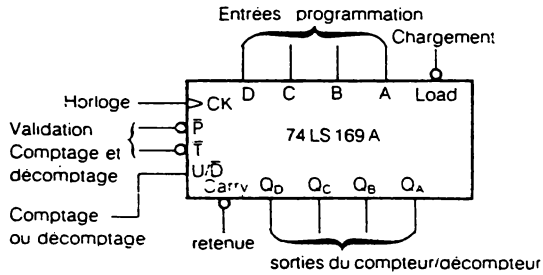


Fig. 12.30. — *Compteur, décompteur 74 LS 169 A.*

$h\uparrow$. $\overline{\text{load}} \rightarrow A$ à D dans Q_A^D

$h\uparrow$. load. $\overline{P}.\overline{T}.\ U/\overline{D} \rightarrow +1$ dans Q_A^D

$h\uparrow$. load. $\overline{P}.\overline{T}.\ \overline{U/\overline{D}} \rightarrow -1$ dans Q_A^D

$h\uparrow$. load. $\overline{\overline{P}.\overline{T}} \rightarrow Q_i$ dans Q_i (recopie du compteur dans lui-même).

$\overline{T}.U/\overline{D}.\ (Q_A^D = 15) + \overline{\overline{T}.U/\overline{D}}.\ (Q_A^D = 0) =$ carry (retenue).

Ce compteur comme le précédent est entièrement synchrone. Il n'a pas de remise à zéro. On peut la faire en chargeant 0 dans le compteur par l'entrée load.

11. LES APPLICATIONS DU COMPTAGE

11.1. Comptage d'événements

Les débutants en logique posent souvent la question : "Des compteur, mais pour compter quoi ?" La question dans sa simplificité même embarasse car les compteurs possèdent de multiples usages, comme nous allons le voir. La plus simple des applications est le comptage d'événements. Ces événements peuvent être des impulsions dans un fréquencemètre, des impulsions provoquées par des phénomènes physiques, etc... Il suffit d'appliquer le signal correspondant à l'entrée comptage d'un compteur.

$E \rightarrow +1$ dans C.

L'événement peut être plus complexe et dépendre de conditions. On élabore alors une fonction booléenne qui est vraie quand l'événement se

produit. On applique cette fonction à l'entrée comptage éventuellement en coïncidence avec une horloge, en prenant soin de respecter les paramètres dynamiques. Ainsi pour mesurer la fréquence d'un signal S, on compte le nombre d'impulsions de S pendant une durée D égale à 1 seconde, ce qui s'exprime par :

$S.D \rightarrow +1$ dans C

11.2. Division de fréquence

Le diagramme des temps d'un compteur progressant en binaire pur (fig. 12.31) montre que si l'horloge a une période T:

- l'étage Q_0 commute avec une période $2T$,
- l'étage Q_1 commute avec une période $4T$,
- l'étage Q_2 commute avec une période $8T$,

Par extension l'étage Q_n commute avec une période $2^n T$. La fréquence F du signal d'horloge est donc divisée par 2^n avec un compteur à n étages.

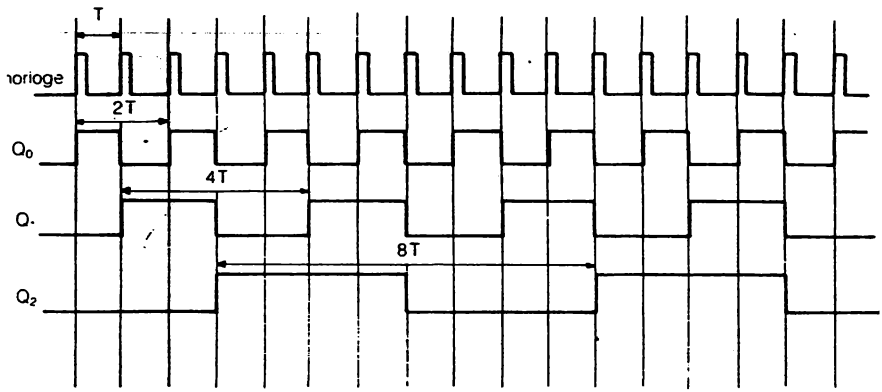


Fig. 12.31. — *Division de fréquence par comptage.*

11.3. Exploitation des états d'un compteur. Parasites de décodage

Dans certaines applications, un compteur compte des événements et certaines positions du compteur servent à exciter un autre circuit séquentiel.

A titre d'exemple prenons le système suivant qui comporte un compteur C, avançant sur une horloge h et une bascule B mise à 1 quand C vaut 0 et mise à 0 quand C vaut 2. Le fonctionnement de ce système se décrit par trois équations :

$h \rightarrow +1$ dans C

$(C = 0) \rightarrow B$ à 1

$(C = 2) \rightarrow B$ à 0.

Les figures 12.32 et 12.33 donnent le diagramme des temps et le schéma correspondant à ces équations. On sait (§ 6.2) qu'il existe des états transitoires en sortie d'un compteur. Ces états transitoires sont détectés par

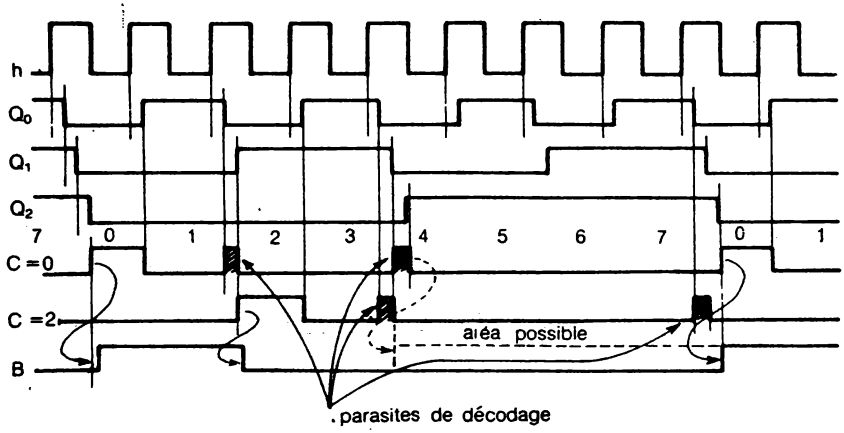


Fig. 12.32. — *Diagramme de décodage d'un compteur.*

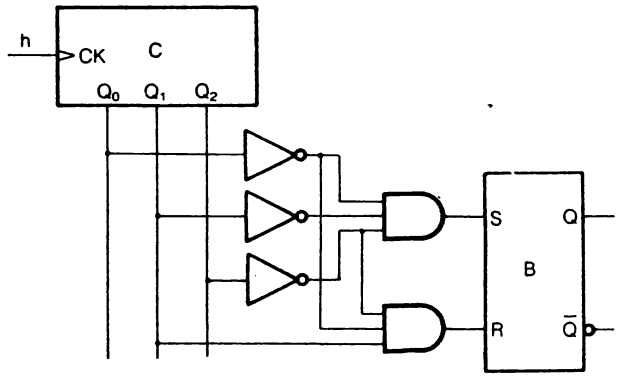


Fig. 12.33. — *Schéma d'exploitation des états d'un compteur conduisant à des aléas.*

le circuit de décodage et peuvent provoquer un basculement indésirable de B si leur durée est suffisante. La figure 12.32 a été dessinée avec un compteur C asynchrone pour lequel les états transitoires sont particulièrement marqués. Les mêmes inconvénients existent pour un compteur synchrone (voir § 6.2).

Pour éviter ces aléas, il faut utiliser les valeurs $C = 0$ et $C = 2$ à un moment où elles sont stables. Les figures 12.34 et 12.35 donnent un exemple avec deux horloges h_1 et h_2 suffisamment décalées : h_1 sert à incrémenter le compteur et h_2 à exciter la bascule en liaison avec les valeurs décodées. Le fonctionnement se décrit par les équations :

$h_1 \rightarrow +1$ dans C

$h_2 . (C = 0) \rightarrow B$ à 1

$h_2 . (C = 2) \rightarrow B$ à 0

La durée T séparant h_1 de h_2 doit être supérieure au retard du compteur et des portes d'inversions, de façon à ce que la valeur à décoder soit stabilisée (voir exercice au paragraphe 8 du chapitre 14).

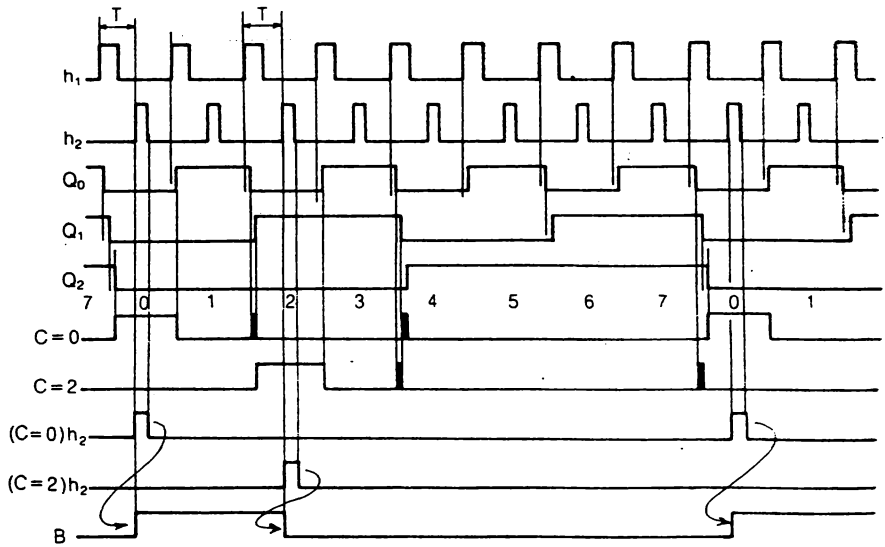


Fig. 12.34. — *Diagramme de décodage d'un compteur avec validation du décodage.*

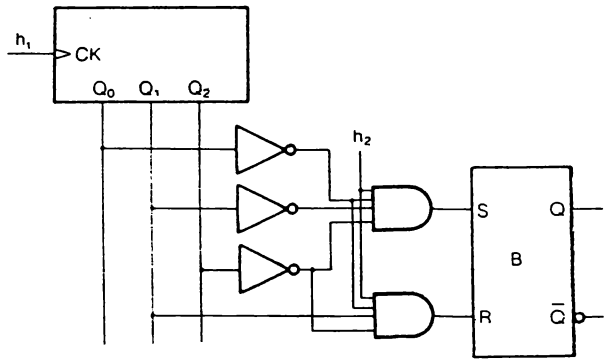


Fig. 12.35. — *Schéma d'exploitation des états d'un compteur avec validation du décodage.*

11.4. Réalisation de cycles d'horloge

Une horloge périodique de période T constitue un cycle à une seule horloge. On l'obtient à partir d'un oscillateur. Un tel signal peut s'avérer insuffisant pour commander certains montages, plusieurs signaux dans la même période T étant nécessaires.

A titre d'exemple le montage du paragraphe 11.3 fonctionnait avec un cycle à deux temps h_1 et h_2. Comment obtenir un tel cycle ou des cycles ayant un nombre d'horloges plus élevé? On procède par division de fréquence (voir § 11.2) et multiplication par l'horloge. La figure 12.36 donne la réalisation d'un cycle à deux temps h_1 et h_2 avec une bascule D travaillant en diviseur par deux.

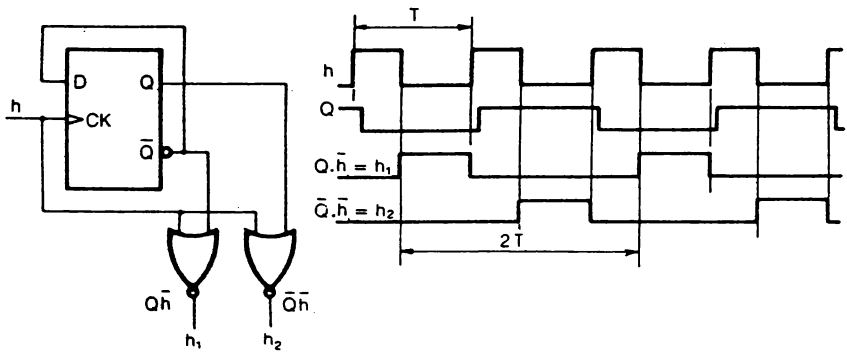
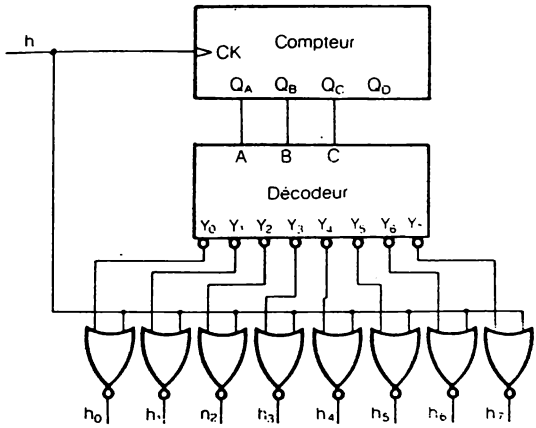
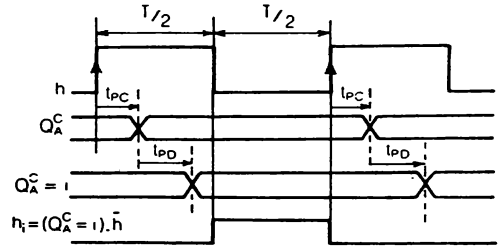


Fig. 12.36. — *Réalisation d'un cycle d'horloge à deux temps.*





t_{PC} : Temps de propagation du compteur
t_{PD} : Temps de propagation du décodeur

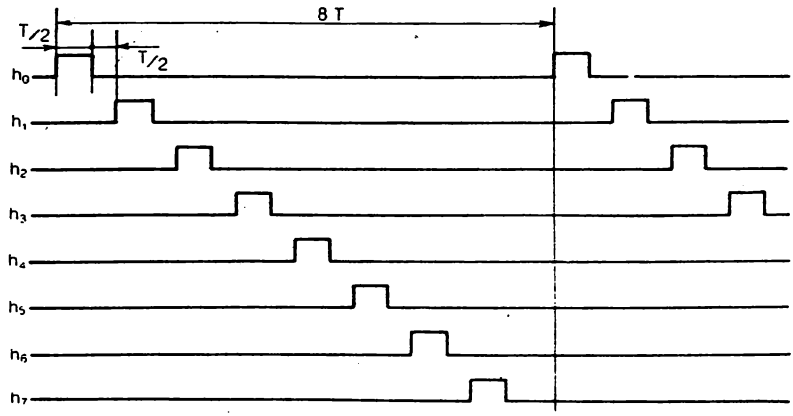


Fig. 12.37. — *Principe de réalisation d'un cycle d'horloge à huit temps.*

Un cycle d'horloge à plus de deux temps s'obtient en divisant l'horloge par un compteur, en décodant les états du compteur par un décodeur, en faisant la multiplication soit par des portes soit par une entrée de validation du décodeur. La figure 12.37 montre le principe de réalisation d'un cycle à huit temps. A partir d'un oscillateur donnant un signal de période T et de durée $T/2$, on obtient huit temps durant chacun $T/2$ et espacés de $T/2$, et se répétant avec une période $8T$ (voir exercice au paragraphe 9 du chapitre 14).

11.5. Réalisation de temporisation

Les périodes des horloges d'un système sont souvent plus courtes que les constantes de temps des phénomènes traités. Pour obtenir des impulsions ou des retards compatibles avec ces constantes de temps, on utilise des compteurs ou des décompteurs progressant sur une horloge rapide. Un décodage à la valeur n permet d'obtenir la durée désirée, égale à nT, T étant la période de l'horloge rapide (fig. 12.38). (Voir le paragraphe 10 du chapitre 14 pour un montage pratique.)

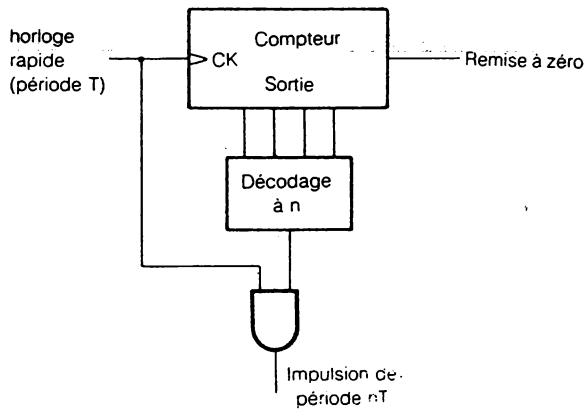


Fig. 12.38. — *Élaboration de temporisation par comptage.*

11.6. Adressage d'éléments binaires

Très souvent des quantités sont rangées séquentiellement, les unes derrière les autres. Ce sont par exemple des groupes d'éléments binaires appelés mots et rangés dans les cases successives d'une mémoire (voir

chapitre 15 sur les mémoires), des groupes d'éléments binaires dans de grands registres à décalage, des éléments binaires transmis en série. Pour repérer chacune de ces entités, on utilise un compteur, dont chaque valeur représente une position.

11.7. Registre de mémorisation

Nous avons vu que les compteurs intégrés possédaient d'autres fonctions que le comptage. Bien souvent ils sont chargeables, d'où l'idée de les utiliser aussi comme simple registre de mémorisation. Cela évite de diversifier les types de boîtiers.

11.8. Rétroactions dans un compteur. Cycle incomplet

Pour réaliser un cycle incomplet nous pouvons calculer un compteur avec des bascules comme indiqué au paragraphe 4 de ce chapitre. Cette méthode bien que rigoureuse est fastidieuse, ne s'applique qu'aux compteurs

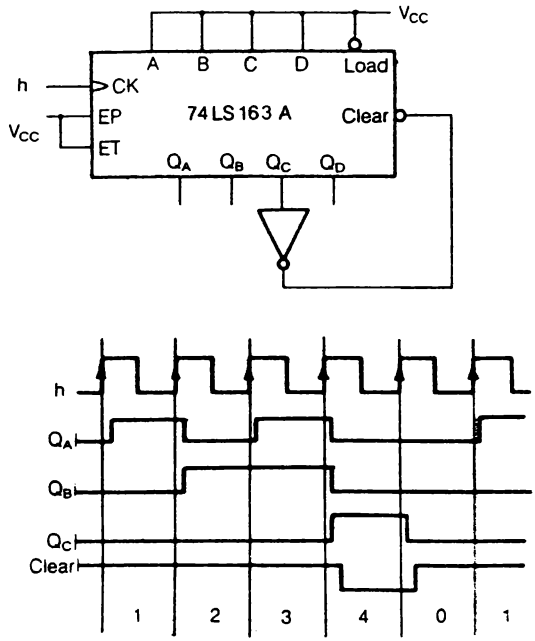


Fig. 12.39. — *Cycle incomplet avec compteur programmable.*

de petite taille et conduit à des montages volumineux car l'on utilise des bascules et des portes, composants très peu denses en intégration (deux bascules par boitiers, quatre portes par boitier). Une autre méthode consiste à réaliser un compteur binaire avec remise à zéro. On décode la dernière valeur du cycle que l'on applique sur l'entrée de remise à zéro prioritaire par rapport au comptage. A la prochaine impulsion d'horloge, au lieu de passer à $N+1$, le compteur passe à 0. Ce procédé se met très facilement en œuvre avec les compteurs programmables intégrés. A titre d'exemple la figure 12.39 montre le cycle 0, 1, 2, 3, 4, 0 ... calculé au paragraphe 4 et réalisé simplement avec un compteur 74 LS 163 A.

Un autre type de cycle, se déroulant entre deux valeurs N_1 et N_2 est réalisable avec un compteur programmable en utilisant l'entrée de chargement load. On décode la valeur finale du cycle N_2 en sortie du compteur. Cette valeur décodée active l'entrée load qui charge la valeur $A\ B\ C\ D$ égale à la valeur initiale du cycle N_1 (fig. 12.40).

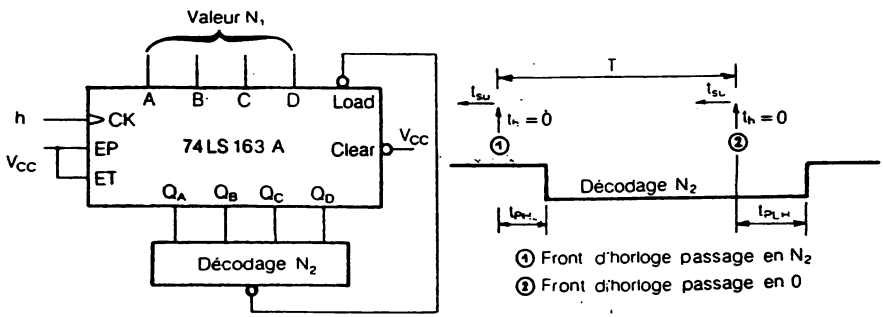


Fig. 12.40. — *Cycle incomplet entre deux valeurs N_1 et N_2.*

Pour fonctionner correctement les rétrocouplages sur les entrées clear et load du compteur doivent respecter essentiellement deux paramètres dynamiques des entrées clear et load : le temps de préaffichage $t_{set\ up}$ et le temps de maintien t_{hold}. Pour le premier cela est facile si la période T entre deux horloges est suffisante ; pour le second cela se réalise automatiquement car le t_{hold} est nul (voir fig. 12.29 sur les paramètres dynamiques du 74 LS 163 A).

On voit parfois des montages à cycle incomplet réalisés avec des compteurs ayant des entrées de programmation ou de remise à zéro asynchrones. Si le compteur doit parcourir le cycle 0 à N, on décode la valeur $N+1$ que l'on applique sur l'entrée de remise à zéro. Cette entrée de remise à zéro

étant asynchrone, agit immédiatement de sorte que la valeur $N+1$ disparaît au profit de la valeur 0. La mise à zéro cesse ainsi et à la prochaine impulsion d'horloge le compteur recommence le cycle 0, 1 ... N. La figure 12.41 montre une réalisation avec un compteur asynchrone 74 LS 93 (remise à zéro asynchrone) et avec un compteur synchrone 74 LS 160A (remise à zéro asynchrone). Ces montages avec entrées asynchrones sont déconseillés pour les raisons suivantes :

- un compteur présente en sortie des états transitoires qui peuvent être interprétés comme étant $N+1$, ce qui provoque une mise à zéro intempestive,
- un compteur, surtout s'il est constitué de plusieurs boîtiers peut avoir des temps de réaction différents pour les différents étages. Certains étages passent rapidement à 0, ce qui supprime la mise à 0, qui peut alors durer un temps insuffisant pour mettre à 0 les étages lents.

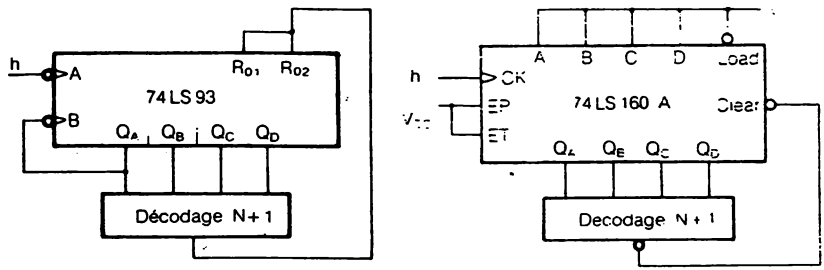


Fig. 12.41. — *Cycles incomplets réalisés avec des compteurs à remise à zéro asynchrone et donnant des aléas de fonctionnement.*

Bien qu'il soit possible de réaliser des cycles incomplets avec des compteurs à entrée de commande asynchrone (il faut ajouter une bascule auxiliaire et un deuxième temps), il est préférable, chaque fois qu'il y a rétroaction, d'employer des circuits à commandes synchrones.

Annexe du chapitre 12

LES COMPTEURS INTÉGRÉS PRINCIPAUX TYPES DE CIRCUITS

1. TABLEAUX DE SYNTHÈSE

Le tableau de la figure 12.42 regroupe les principaux compteurs intégrés que l'on trouve dans les catalogues pour la famille LS. Nous les avons classés en fonction de quelques critères principaux dégagés au chapitre 12.

Deux grandes familles peuvent être distinguées, les compteurs asynchrones et les compteur synchrones.

2. LES COMPTEURS ASYNCHRONES

2.1. Les compteurs non affichables (74 LS 90 – 74 LS 92 – 74 LS 93 – 74 LS 290 – 74 LS 293)

A la mise à 9 et au brochage des alimentations près, ces différents boîtiers de compteurs se caractérisent par le schéma de boîtier de la figure 12.43.

Type de boîtier	Nombres de broches	Commandes communes							Mémorisation				Sorties				Observations
		Comptage							Type de comptage								
		Mode	Fréquence max MHz	Commande inhibition	Commande décomptage	Mise à zéro (Clear)	Mise à neuf	Affichage	Type de bascule	Action	Nombres d'étages	diviseur par	Q	retenue	Temps de propagation ns	Ampli sur sorties	
74 LS 90	14	A	32			A	A		J K	↓	4	10.5.2	X				brochage de l'
74 LS 92	14	A	32			A			J K	↓	4	12.6.2	X		50		alimentation
74 LS 93	14	A	32			A			J K	↓	4	16.8.2	X		70		non normalisé
74 LS 290	14	A	32			A	A		J K	↓	4	10.5.2	X		50		Idem 90 et 93
74 LS 293	14	A	32			A			J K	↓	4	16.8.2	X		70		avec brochage d'alimentation normalisé
74 LS 196	14	A	30			A			JK-T	↓	4	10.5.2	X		45		compteurs affichables
74 LS 197	14	A	30			A		A	T	↓	4	16.8.2	X		95		
74 LS 390	16	A	25			A			T	↓	4	10.5.2	X		60	X	doubles compteurs
74 LS 393	14	A	25			A			T	↓	4	16	X		60	X	
74 LS 490	16	A	25			A	A		T	↓	4	10	X		54	X	
74 LS 160 A	16	S	25	S		A		S	D	↑	4	10	X	X	35		Clear asynchrone
74 LS 161 A	16	S	25	S		A		S	D	↑	4	16	X	X	35		
74 LS 162 A	16	S	25	S		S		S	D	↑	4	10	X	X			Clear synchrone
74 LS 163 A	16	S	25	S		S		S	D	↑	4	16	X	X			
74 LS 190	16	S	20	S	S			A	J K	↑	4	10	X	X			
74 LS 191	16	S	20	S	S			A	J K	↑	4	16	X	X			
74 LS 192	16	S	25			A		A	T	↑	4	10	X	X			entrées et retenues séparées de comptage et décomptage
74 LS 193	16	S	25			A		A	T	↑	4	16	X	X			
74 LS 168 A	16	S	25	S	S			S	D	↑	4	10	X	X			
74 LS 169 A	16	S	25	S	S			S	D	↑	4	16	X	X			

Bascule T variante de bascule RS synchrone (ou RST).

Pour les compteurs asynchrones t_{PLH} ou t_{PHL} en

A : Commande asynchrone – S : Commande synchrone.

Fig. 12.42. — *Les compteurs intégrés.*

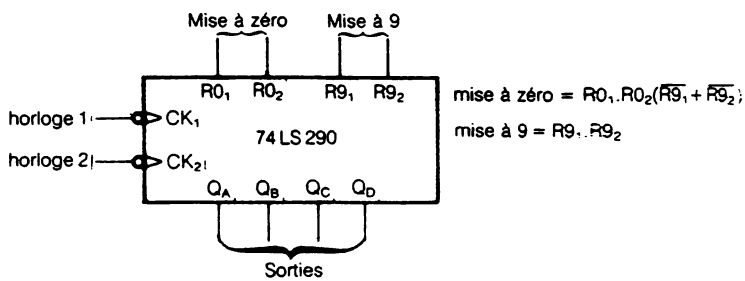


Fig. 12.43. — *Schéma de boîtier du compteur 74 LS 290.*

2.2. Les compteurs affichables (74 LS 196 – 74 LS 197)

Ils diffèrent des précédents par une commande d'affichage (fig. 12.44).

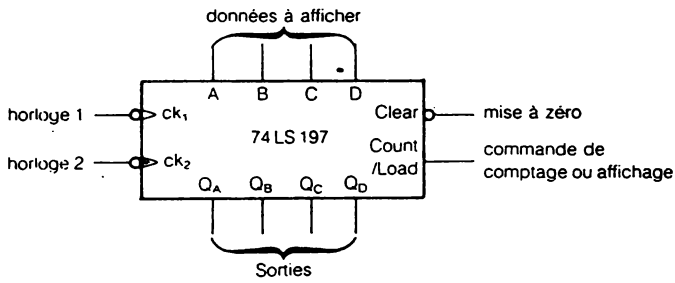


Fig. 12.44. – *Schéma de boîtier du compteur 74 LS 197.*

2.3. Les doubles compteurs (74 LS 390 – 74 LS 393 – 74 LS 490)

Ils se caractérisent par l'existence de deux compteurs indépendants non affichables à quatre étages par boîtier, le compteur 74 LS 393 donné à la figure 12.45 illustre cette famille.

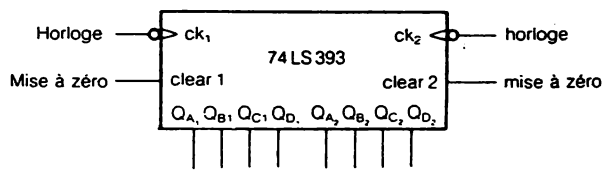


Fig. 12.45. – *Schéma de boîtier du compteur 74 LS 393.*

3. LES COMPTEURS SYNCHRONES

Ils sont tous affichables mais n'ont pas nécessairement une remise à zéro.

3.1. Les compteurs synchrones non reversibles (74 LS 160 A – 74 LS 161 A – 74 LS 162 A – 74 LS 163 A)

Ils existent en versions avec mise à zéro asynchrone et mise à zéro synchrone (fig. 12.46).

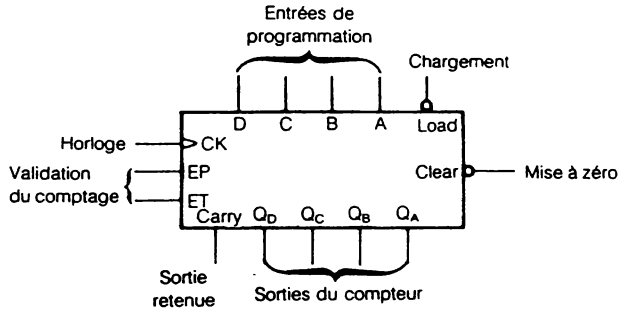


Fig. 12.46. – *Schéma de boîtier des compteurs (74 LS 160 A – 74 LS 161 A – 74 LS 162 A – 74 LS 163 A).*

3.2. Les compteurs synchrones réversibles (74 LS 190 – 74 LS 191 – 74 LS 192 – 74 LS 193 – 74 LS 168 A – 74 LS 169 A)

Il n'existe pas de décompteurs intégrés seuls. On trouve dans le même circuit les fonctions de comptage et de décomptage, d'où le nom de compteurs réversibles. Les circuits passent du comptage au décomptage sous l'action d'une commande.

3.2.1. Compteurs synchrones réversibles à mise à zéro asynchrone (74 LS 190 – 74 LS 191)

Le schéma de boîtier de ces compteurs est donné par la figure 12.47.

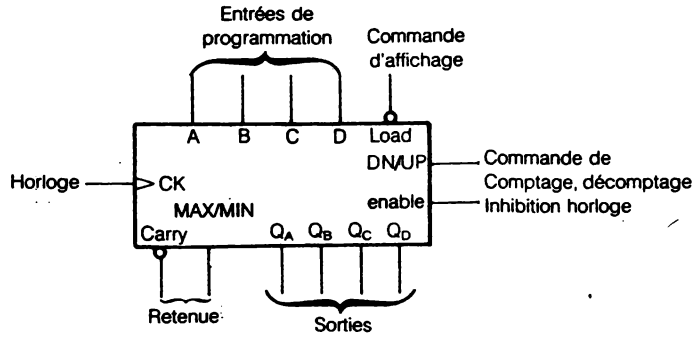


Fig. 12.47. — *Schéma de boîtier des compteurs (74 LS 190 — 74 LS 191).*

3.2.2. Compteurs synchrones réversibles avec chargement synchrone

(74 LS 168 A — 74 LS 169 A)

Le schéma de boîtier de ces compteurs est donné à la figure 12.48.

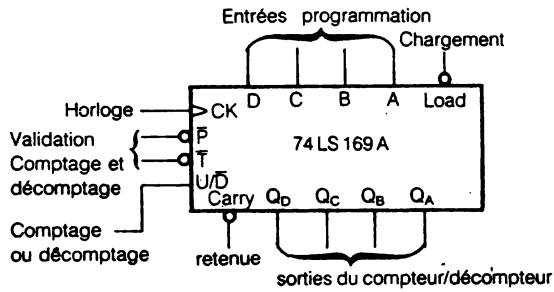


Fig. 12.48. — *Schéma de boîtier des compteurs (74 LS 168 A — 74 LS 169 A).*

3.2.3. Compteurs synchrones réversibles à mise à zéro et affichage

Alors que les deux types de compteurs précédents utilisent une horloge commune et une commande d'état qui indique le sens de progression, ce compteur utilise deux horloges différentes et a deux retenues (fig. 12.49). Moins commode d'emploi, il correspond à une première génération de compteurs réversibles.

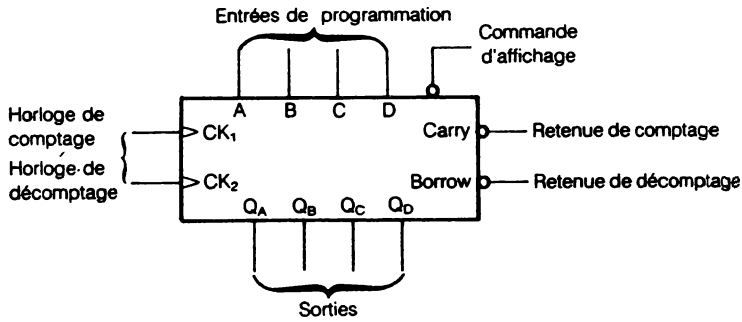


Fig. 12.49. — *Schéma de boîtier des compteurs 74 LS 192 — 74 LS 193.*

4. CHOIX D'UN COMPTEUR

Les compteurs sont très nombreux. Le choix d'un type de compteur dépend surtout de la manière dont il est utilisé. Voici quelques critères :

- si les excitations du compteur sont simples : commande de progression et mise à zéro seulement, un compteur asynchrone suffit généralement ;
- si les excitations sont variées ; progression, arrêt, programmation, remise à zéro on s'oriente vers un compteur synchrone ;
- si le compteur s'automodifie par des chargements ou des remises à zéro, il faut prendre un compteur synchrone à commandes auxiliaires synchrones ;
- les caractéristiques du compteur (fréquence maximum, durée minimum des impulsions, paramètres de $t_{set\ up}$, t_{hold} doivent être compatibles avec les caractéristiques des événements appliqués au compteur.

Dans le cas de commandes et actions variées au niveau du compteur le meilleur choix est d'utiliser les compteurs synchrones à commande synchrone

- pour les compteurs : 74 LS 162 A pour les décades
  74 LS 163 A pour les diviseurs par 16.
- pour les compteurs décompteurs : 74 LS 168 A pour les décades
  74 LS 169 A pour les diviseurs par 16.

Chapitre 13

LA FONCTION DE DÉCALAGE

1. DÉFINITION DE LA FONCTION DE DÉCALAGE

Il s'agit de réaliser un registre constitué de bascules Q_0, Q_1 ... Q_i ... Q_n, commandé par une horloge, comprenant des liaisons entre les bascules telles qu'après chaque impulsion d'horloge, la configuration contenue dans le registre ait glissé d'une position. Ce qui s'écrit :

$h \rightarrow Q_i$ dans Q_{i+1}

L'entrée de Q_0 qui n'a pas d'antécédent s'appelle entrée série, tandis que la sortie Q_n qui n'a pas de successeur s'appelle sortie série.

L'équation générale devient :

$h \rightarrow ES$ dans Q_0

$h \rightarrow Q_0$ dans Q_1

$h \rightarrow Q_1$ dans Q_2

|

|

|

$h \rightarrow Q_{n-1}$ dans Q_n

$Q_n \rightarrow$ sortie série.

Ces équations montrent qu'il y a des rétroactions croisées, (l'état d'une bascule sert à l'affichage de la suivante, qui, elle-même, alimente la suivante et ainsi de suite). Nous savons d'après l'analyse faite au chapitre 10 qu'il faut utiliser des bascules. La figure 13.1 montre des exemples de registres à décalage.

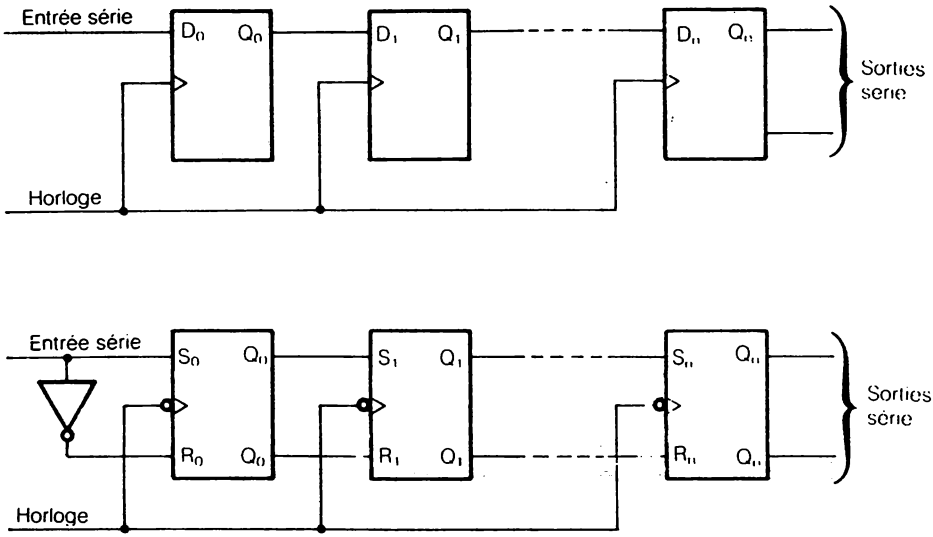


Fig. 13.1. — *Exemple de registres à décalage.*

2. LES DIFFÉRENTS TYPES DE REGISTRES A DÉCALAGE

2.1. Registre à entrée série et sortie série (fig. 13.2a)

L'information se propage dans le registre sans que les étages intermédiaires soient accessibles. Cette structure sert à réaliser des mémoires organisées en file (mémoire FIFO à 64 étages par exemple).

2.2. Entrée série, sortie parallèle (fig. 13.2b)

Dans cette structure tous les étages sont accessibles. Cette organisation

convient aux registres de taille moyenne (4 ou 8 eb), compte tenu du nombre réduit des broches des circuits intégrés.

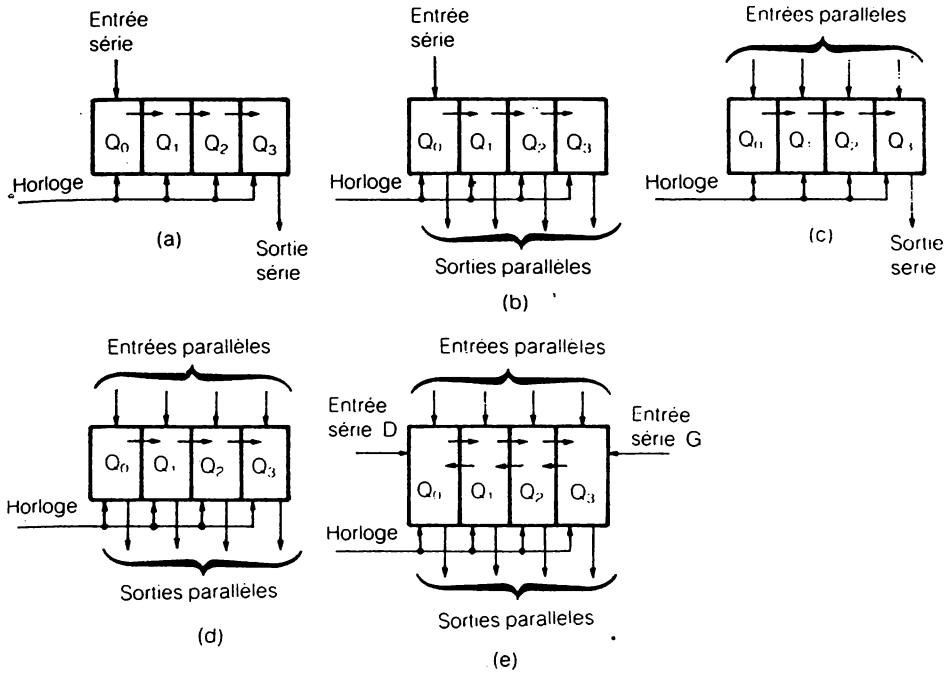


Fig. 13.2. — *Différents types de registres à décalage.*

2.3. Entrée parallèle, sortie série (fig. 13.2c)

Les étages sont affichés simultanément par une commande, seul le dernier étage est accessible en lecture.

2.4. Entrée parallèle, sortie parallèle (fig. 13.2d)

On affiche tous les étages en parallèle et ils sont accessibles en parallèle en sortie. Ce type d'organisation convient pour les registres à quatre ou huit éléments binaires.

2.5. Registre bidirectionnel (fig. 13.2e)

Il existe aussi des registres qui permettent le décalage à droite ou à gauche, décalage combiné avec les possibilités d'entrée et sorties séries ou parallèles. Les tailles sont de 4 ou de 8 eb.

3. LES DIFFÉRENTS TYPES DE DÉCALAGES UTILISÉS

Les machines de traitement de l'information utilisent plusieurs types de décalage.

3.1. Décalage logique à droite ou à gauche

C'est le décalage classique où l'on décale tous les étages de la même façon, avec un 0 ou un 1 entrant dans le registre, l'élément binaire sortant étant soit perdu, soit sauvegardé dans une bascule pour exploitation ultérieure (fig. 13.3).

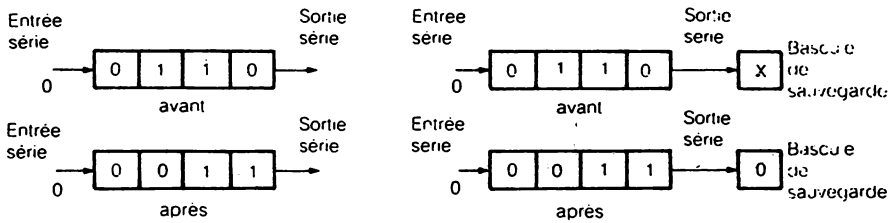


Fig. 13.3. — *Décalage logique.*

3.2. Décalage circulaire (rotation) à droite ou à gauche

L'eb sortant du registre sur la sortie série est réinjecté à l'entrée série. De cette manière, l'information se conserve (fig. 13.4).

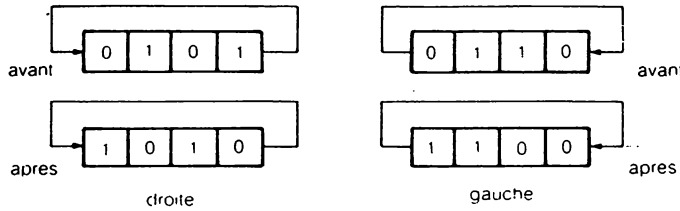


Fig. 13.4. — *Décalage circulaire droite et gauche.*

3.3. Décalage numérique (ou arithmétique)

Nous avons vu dans le chapitre 3 que la multiplication ou la division par des puissances de 2 revenait à décaler les nombres à gauche ou à droite. Pour des nombres non signés, on emploie le décalage logique avec un zéro entrant. Pour des nombres signés en complément à 2, on doit conserver et propager l'eb de signe (voir § 7, chap. 3). Si un bloc de calcul utilise des décalages variés, il faut réaliser les différents aiguillages nécessaires autour des eb d'extrémité du registre à décalage.

4. PRÉLÈVEMENT D'INFORMATION DANS UN REGISTRE A DÉCALAGE

Lors de l'étude des compteurs, nous avions examiné la façon de prélever les états d'un compteur alors qu'il était en train d'évoluer. Nous avions vu que l'on pouvait lire son état à condition d'utiliser une horloge différente de celle qui l'incrémentait. Les mêmes précautions sont à prendre pour un registre à décalage (fig. 13.5). (respect des conditions de t_{PLH} et t_{PHL} des bascules constituant le registre).

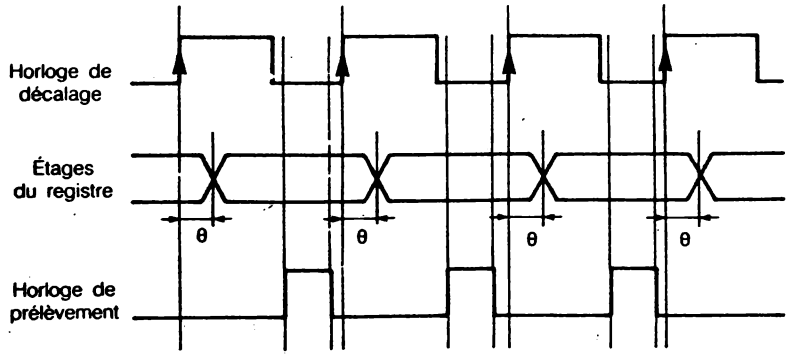


Fig. 13.5. — *Prélèvement dans un registre à décalage.*

5. PRINCIPALES CARACTÉRISTIQUES DES REGISTRES INTÉGRÉS

5.1. La fonction horloge

L'horloge agit, soit sur le front montant, soit sur le front descendant. Quelques circuits possèdent une fonction d'inhibition de l'horloge qui pour certains n'est pas synchrone (74 LS 165, 74 LS 166, 74 LS 95 B), elle agit par l'intermédiaire d'une porte sur l'entrée *CK* des bascules. Nous avons déjà signalé les particularités d'un tel montage (voir § 3, chap. 11 sur les registres de mémorisation). Ceci se retrouve aussi sur certains compteurs anciens. Les registres récents ont des entrées de commande synchrone, ce qui permet d'agir sur les entrées *D* (ou *RS*) et de laisser l'horloge indépendante (fully independant clock).

5.2. Les commandes

Les différentes commandes que l'on peut trouver au niveau des registres à décalage intégrés sont le décalage, le décalage droite, le décalage gauche, la mise à zéro du registre, l'inhibition de l'horloge, l'affichage d'un nouveau contenu, la commande trois états des sorties.

Dans les circuits les plus complexes, pour réduire le nombre de broches, une partie des commandes est codée sur deux entrées "mode

control" qui permettent de programmer quatre modes de fonctionnement, par exemple: décalage droite, décalage gauche, affichage et inhibition horloge. Toutes les commandes, sauf à la mise à zéro, sont synchrones.

5.3. Les entrées sorties

5.3.1. L'entrée série

Elle peut prendre plusieurs formes:

- Une entrée unique comme à la figure 1.
- Une porte pour permettre des combinaisons à l'entrée.
- Un double accès $J\overline{K}$ pour faire travailler le premier étage comme une bascule JK (maintien, affichage en 1 ou 0, commutation).

5.3.2. La sortie série

Elle ne présente pas de particularité, c'est l'état Q_n du dernier étage du registre à décalage.

5.3.3. Les entrées parallèles

Elles permettent de charger dans le registre une nouvelle valeur, elles sont sous la dépendance de la commande d'affichage (entrée indépendante ou codée par une combinaison "mode control").

5.3.4. Les sorties parallèles

Elles permettent d'accéder aux différents étages du registre.

Certains registres disposent de sorties parallèles sous la dépendance d'une commande 3 états. Pour d'autres circuits, les entrées-sorties communes sont également sous la dépendance d'une commande trois états.

Cette dernière particularité, outre le fait qu'elle économise les broches d'entrée-sortie au niveau du boîtier, permet une exploitation adaptée à une architecture en bus.

Dans le cas d'une commande trois états, le dernier étage du registre est présenté en sortie sous la dépendance de cette commande mais également

sur une autre sortie directement pour assurer la fonction de décalage série dans le cas d'association de boîtiers.

5.4. Organisation interne d'un registre à décalage

La figure 13.6 donne le schéma de principe d'un registre à décalage universel réalisé avec des bascules *D*, à entrées-sorties communes, qui illustre les différentes commandes et fonctions que nous venons d'examiner.

Un bloc combinatoire interne calcule les différentes commandes à partir des fonctions présentées aux entrées du boîtier. On remarque l'intérêt des circuits 3 états qui permettent d'effectuer simplement l'aiguillage d'entrée-sortie.

L'horloge étant appliquée en permanence et directement sur l'entrée *CK* ("fully independant clock"), les différentes fonctions sont obtenues en

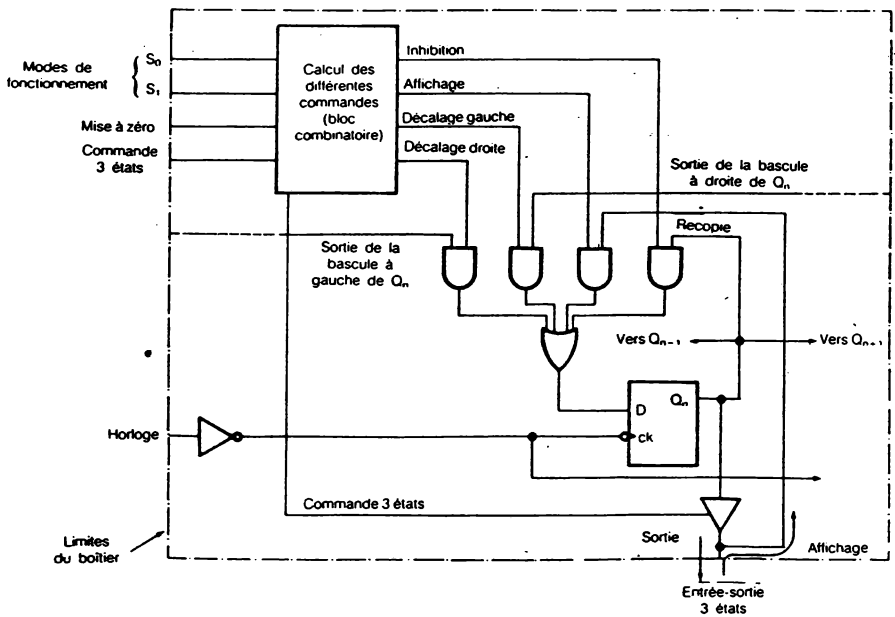


Fig. 13.6 — *Registre à décalage universel : principe.*

agissant sur l'entrée D avec une information convenable: état de la bascule de rang $n-1$ ou $n+1$ pour les décalages droite ou gauche, état de la bascule elle-même (rang n) dans le cas de l'inhibition horloge (recopie de la bascule dans elle-même) et blocage de toutes les commandes dans le cas de la mise à zéro (présentation d'un zéro sur l'entrée D de chaque étage).

6. EXEMPLES DE REGISTRES INTÉGRÉS

6.1. Registre à 4 eb 74 LS 194

6.1.1. Principe et fonctionnement

La figure 13.7 montre un exemple de registre intégré 74 LS 194. Il comprend:

- Une entrée horloge.
- Une entrée de mise à zéro clear, qui met les quatre étages Q_A, Q_B, Q_C, Q_D à 0.
- Deux entrées de commande S_0 et S_1 qui déterminent le mode de fonctionnement du registre, sous la dépendance de l'horloge.
  $\overline{S}_1\ \overline{S}_0 \rightarrow$ le registre conserve son état
  $\overline{S}_1\ S_0 \rightarrow$ le registre est décalé à droite
  $S_1\ \overline{S}_0 \rightarrow$ le registre est décalé à gauche
  $S_1\ S_0 \rightarrow$ le registre est chargé avec une valeur
- Quatre entrées A, B, C, D qui servent au chargement.
- Une entrée série L pour le décalage à gauche.
- Une entrée série R pour le décalage à droite.
- Quatre sorties Q_A, Q_B, Q_C, Q_D.

Ce registre à décalage est tel que:

- la mise à zéro est prioritaire sur toutes les autres commandes et s'effectue indépendamment de l'horloge (mise à zéro asynchrone).
- Les commandes liées à l'état de $S_1\ S_0$ sont synchrones de l'horloge sur un front montant (positive edge triggered).

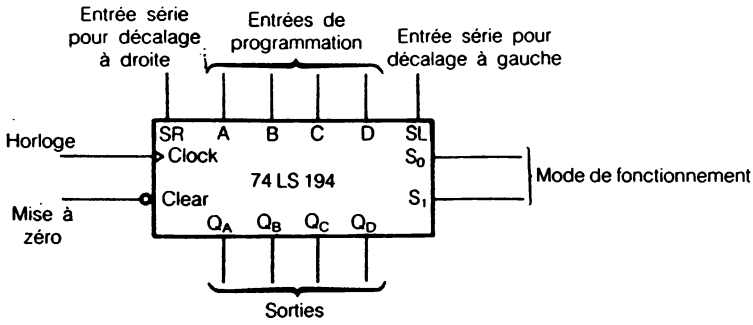


Fig. 13.7. — *Registre à décalage bidirectionnel 74 LS 194.*

On peut décrire le fonctionnement de ce registre par un tableau de vérité ou par des équations logiques. Ce deuxième mode présente l'avantage, dans ce cas, d'être plus "parlant".

$\overline{\text{Clear}}$ $\rightarrow 0$ dans Q_A^D

Clear . $\overline{S}_1 \overline{S}_0 \rightarrow Q_A^D$ dans Q_A^D (maintien de l'état)

Clear.$h\uparrow$.$\overline{S}_1 S_0 \rightarrow R$ dans Q_A, Q_A dans Q_B, Q_B dans Q_C, Q_C dans Q_D (décalage à droite)

Clear.$h\uparrow$.$S_1 \overline{S}_0 \rightarrow L$ dans Q_D, Q_D dans Q_C, Q_C dans Q_B, Q_B dans Q_A (décalage à gauche)

Clear.$h\uparrow$.$S_1 S_0 \rightarrow A$ à D dans Q_A^D (chargement)

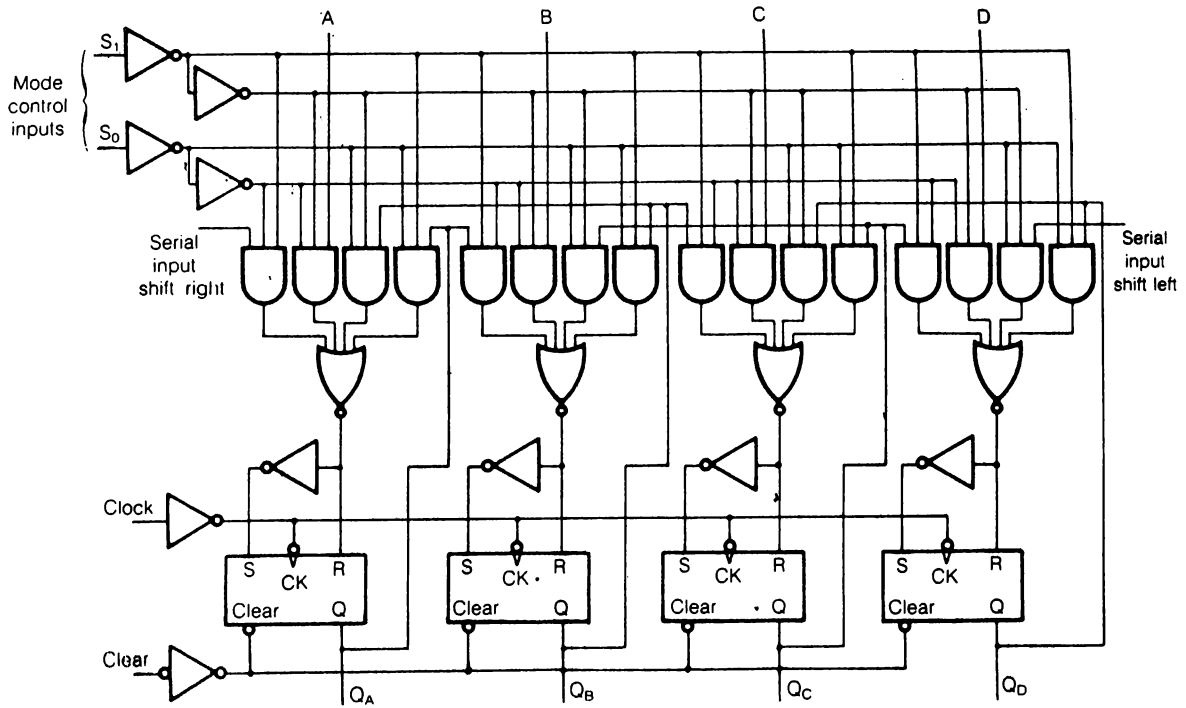
La figure 13.8 donne le schéma de détail de ce registre. Il comprend quatre bascules *RS* synchrones, avec une horloge commune distribuée directement sur les quatre entrées horloge sans qu'elle soit combinée avec d'autres entrées. L'action du clear est directe et les fonctions liées au mode (S_1, S_0) sont réalisées par un réseau combinatoire.

A titre d'exemple, on vérifie sur l'étage Q_B les quatre modes de fonctionnement donnés plus haut:

$$S_B = \underset{\substack{\uparrow \\ \text{Décalage} \\ \text{à droite}}}{Q_A \overline{S}_1 S_0} + \underset{\substack{\uparrow \\ \text{Chargement}}}{B S_1 S_0} + \underset{\substack{\uparrow \\ \text{Décalage} \\ \text{à gauche}}}{Q_C S_1 \overline{S}_0} + \underset{\substack{\uparrow \\ \text{Maintien}}}{Q_B \overline{S}_1 \overline{S}_0}$$

$$R_B = \overline{S}_B$$

ce qui correspond bien aux équations données plus haut.



Clear : mise à zéro.
Serial input shift right : entrée série pour décalage à droite.
Serial input shift left : entrée série pour décalage à gauche
Mode control inputs : entrées de mode de fonctionnement
A B C D : entrées parallèles
$Q_AQ_BQ_CQ_D$: sorties parallèles

Fig. 13.8. — *Schéma bloc du registre 74 LS 194 A.*

6.1.2. Paramètres électriques et dynamiques du 74 LS 194

On retrouve les paramètres habituels (fig. 13.9 et 13.10). On note en particulier le t_{hold} nul qui permet la mise en cascade sans aucun problème.

Paramètres	MIN	NOM	MAX	Unité
Tension d'alimentation V_{CC}	4,75	5	5,25	V
Courant de sortie, niveau haut I_{OH}			−400	µA
Courant de sortie, niveau bas I_{OL}			8	mA
Température de fonctionnement T_A	0		70	°C
Tension d'entrée, niveau haut V_{IH}	2			V
Tension d'entrée, niveau bas V_{IL}			0,8	V
Tension d'écrétage V_{IK}			−1,5	V
Tension de sortie niveau haut V_{OH}	2,7	3,5		V
Tension de sortie niveau bas V_{OL}		0,35	0,5	V
Courant d'entrée à la tension d'entrée maximum			0,1	mA
Courant d'entrée niveau haut I_{IH}			20	µA
Courant d'entrée niveau bas I_{IL}			−0,4	mA
Courant de court-circuit en sortie I_{OS}			−100	mA
Courant d'alimentation		15	23	mA

Fig. 13.9. — *Paramètres électriques du registre à décalage 74 LS 194.*

6.2. Registre à 8 eb 74 LS 299 à entrées-sorties communes

Étendre à 8 eb, un registre comme le 74 LS 194 étudié au paragraphe précédent conduirait à passer de 16 à 24 broches. Les constructeurs proposent une version à 20 broches seulement en plaçant les entrées et sorties parallèles sur les mêmes broches (entrées-sorties bidirectionnelles). La figure 13.11 montre le schéma bloc d'un tel registre. Il y a :

— Une entrée de remise à zéro asynchrone Clear.

Paramètres		MIN	NOM	MAX	Unité
Fréquence de l'horloge f_{clock}		0		25	MHz
Durée de l'horloge ou de clear t_w		20			nS
Temps de préaffichage t_{su}	Entrees S_0,S_1	30			nS
	Entrees séries et paralleles	20			nS
	Clear etat inactif	25			nS
Temps de maintien pour toutes les entrées t_h		0			nS
t_{PHL} depuis l'entrée Clear			19	30	nS
t_{PLH} depuis l'horloge			14	22	nS
t_{PHL} depuis l'horloge			17	25	nS

Fig. 13.10. — *Paramètres dynamiques du registre à décalage 74 LS 194.*

— Une entrée horloge.

— Deux entrées S_1 S_0 donnant le mode de fonctionnement de la même manière que pour le 74 LS 194.

— Deux entrées série *SR* et *SL* pour les décalages à droite et à gauche.

— Deux entrées G_1 et G_2 servant à valider les entrées/sorties en basse impédance.

— Deux sorties Q'_A et Q'_H donnant l'état des deux eb d'extrémité du registre.

— 8 entrées/sorties bidirectionnelles A/Q_A, B/Q_B ... H/Q_H pour entrer ou sortir l'information en parallèle.

Comme précédemment, on peut décrire le fonctionnement de ce registre par son tableau de vérité ou par des équations logiques déduites de ce tableau.

$\overline{\text{Clear}} \rightarrow 0$ dans Q_A^H (clear asynchrone prioritaire sur les autres commandes

$G_1 + G_2 + S_1 S_0 \rightarrow A/Q_A$ à H/Q_H à l'état haute impédance

$\overline{G}_1 \overline{G}_2 . \overline{S_1 S_0} \rightarrow A/Q_A$ à H/Q_H à l'état basse impédance

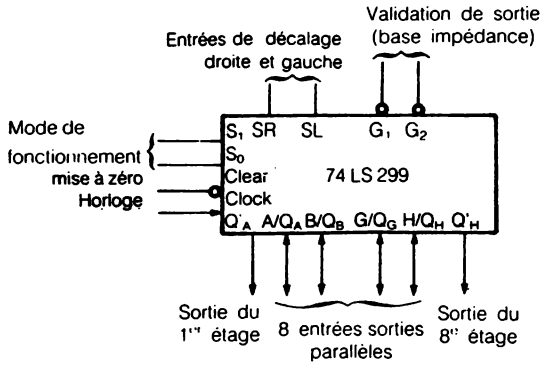


Fig. 13.11. — *Registre à décalage à entrées-sorties communes.*

$\overline{S}_1\,\overline{S}_0$ → Q_A^H dans Q_A^H (maintien indépendant de l'horloge)

$h\uparrow.\,\overline{S}_1\,S_0$ → SR dans Q_A, Q_A dans Q_B, ... Q_G dans Q_H (décalage à droite)

$h\uparrow.\,S_1\,\overline{S}_0$ → SL dans Q_H, Q_H dans Q_G, ... Q_B dans Q_A (décalage à gauche)

$h\uparrow.\,S_1\,S_0$ → $A, B, \ldots G, H$, dans Q_A^H (chargement)

La commande $S_1\,S_0$ met les sorties Q_A à Q_H en mode haute impédance de façon à permettre l'affichage des entrées A, B, ... G, H sur les mêmes broches.

Ce registre est un peu plus difficile à employer que le 74 LS 194 car il faut tenir compte des entrées-sorties communes qui doivent être gérées par des commandes trois états.

6.3. Le décalage combinatoire

Bien souvent, on a besoin de décaler une information de n pas. Pour cela, on charge l'information initiale dans un registre, on met un compteur à 0, et l'on effectue ensuite n opérations de décalage, comptage et test du compteur comme le montre la figure 13.12. La nature série de ce procédé et la fréquence maximum de décalage et de comptage limitent la vitesse de décalage.

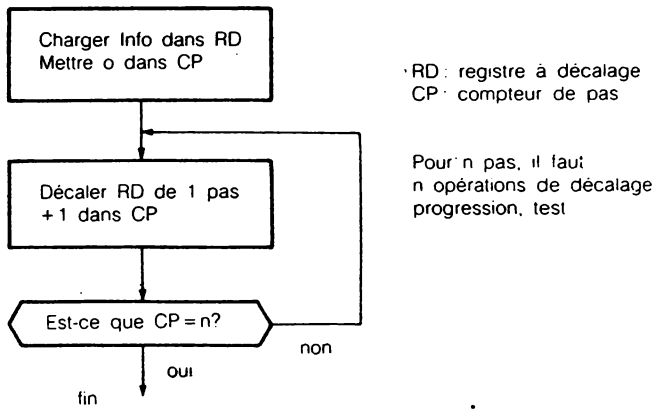
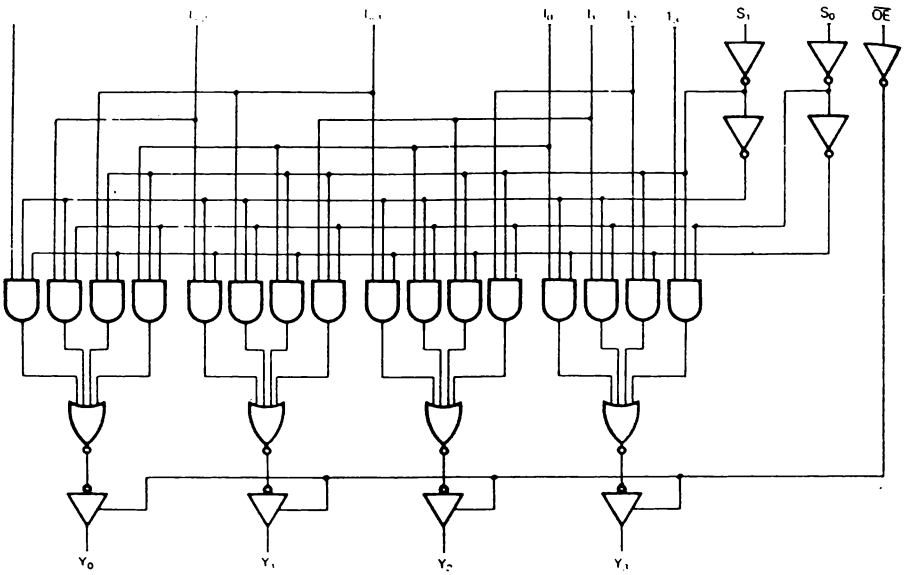


Fig. 13.12. — *Décalage de n pas par un procédé séquentiel ($n \neq 0$).*

On peut accélérer cette opération en réalisant le décalage de manière combinatoire. La figure 13.13 montre le schéma bloc et le tableau de vérité d'un décaleur combinatoire Am 25 S10, proposé par Advanced Micro Devices.

Le nombre de pas de décalage est directement fourni par les deux commandes $S_1 S_0$. $S_1 S_0 = 00$ laisse l'information inchangée, $S_1 S_0 = 01$ la décale de un pas, $S_1 S_0 = 2$ de deux pas et $S_1 S_0 = 3$ de trois pas. Les informations $I_3 I_2 I_1 I_0$ représentent les 4 eb du décaleur au repos (pas de décalage), tandis que I_{-1}, I_{-2} et I_{-3} représentent l'information précédant $I_3 I_2 I_1 I_0$, utilisée pour les décalages de 1, 2 et 3 pas. Les sorties Y du décaleur sont décrites par les quatre équations suivantes :



OE	S_1	S_0	I_3	I_2	I_1	I_0	I_{-1}	I_{-2}	I_{-3}	Y_3	Y_2	Y_1	Y_0
1	X	X	X	X	X	X	X	X	X	Z	Z	Z	Z
0	0	0	D_3	D_2	D_1	D_0	X	X	X	D_3	D_2	D_1	D_0
0	0	1	X	D_2	D_1	D_0	D_{-1}	X	X	D_2	D_1	D_3	D_{-1}
0	1	0	X	X	D_1	D_0	D_{-1}	D_{-2}	X	D_1	D_0	D_{-1}	D_{-2}
0	1	1	X	X	X	D_0	D_{-1}	D_{-2}	D_{-3}	D_0	D_{-1}	D_{-2}	D_{-3}

Fig. 13.13. — *Décaleur combinatoire Am 25 S10.*

$$Y_0 = \overline{S}_1 \overline{S}_0 I_0 + \overline{S}_1 S_0 I_{-1} + S_1 \overline{S}_0 I_{-2} + S_1 S_0 I_{-3}$$
$$Y_1 = \overline{S}_1 \overline{S}_0 I_1 + \overline{S}_1 S_0 I_0 + S_1 \overline{S}_0 I_{-1} + S_1 S_0 I_{-2}$$
$$Y_2 = \overline{S}_1 \overline{S}_0 I_2 + \overline{S}_1 S_0 I_1 + S_1 \overline{S}_0 I_0 + S_1 S_0 I_{-1}$$
$$Y_3 = \overline{S}_1 \overline{S}_0 I_3 + \overline{S}_1 S_0 I_2 + S_1 \overline{S}_0 I_1 + S_1 S_0 I_0$$

Pas de décalage — Décalage de 1 pas — Décalage de 2 pas — Décalage de 3 pas

Avec ce circuit, la durée du décalage ne dépend plus d'une boucle séquentielle répétant les mêmes opérations jusqu'à ce que l'on atteigne le nombre de pas, mais seulement du retard de quelques couches logiques.

Il existe aussi des circuits décaleurs combinatoires internes à des blocs de calcul complexes, qui permettent de décaler l'information d'une position à droite ou à gauche avant de la mémoriser dans un registre interne (*Exemple*: bloc de calcul 74 S 281 ou Am 2901).

7. APPLICATION DU DÉCALAGE

7.1. Division de fréquence

Nous avons vu au chapitre 12 qu'un compteur à n étages permettait de réaliser une division de fréquence par 2^n. Avec un registre à décalage de n étages dans lequel circule en décalage circulaire, un seul 1 parmi des 0, on obtient une division par n. La figure 13.14 montre le cas où $n = 3$.

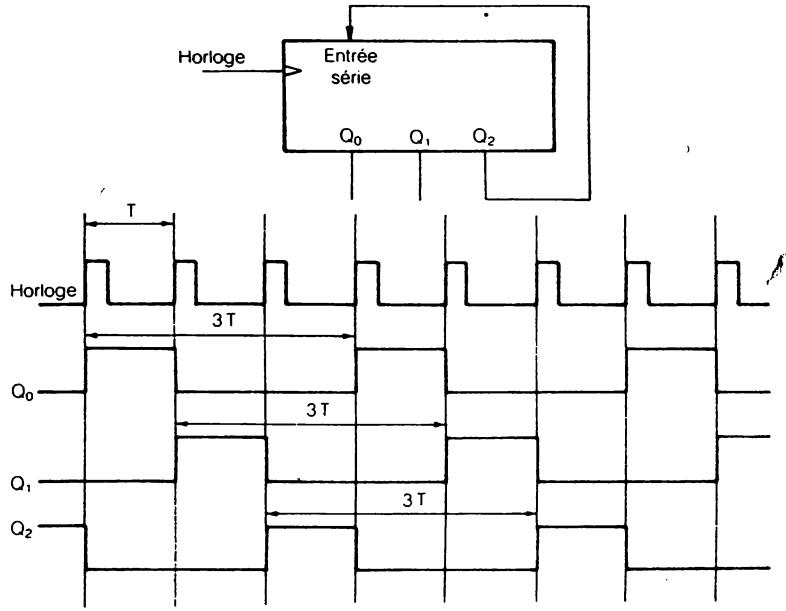


Fig. 13.14. — *Division de fréquence par registre à décalage.*

7.2. Traitement série de l'information

Lorsque l'information est transmise ou reçue en série, on peut profiter de son passage dans le registre à décalage pour effectuer des traitements, par exemple calculer des éléments binaires de redondance qui seront ajoutés à l'information transmise ou vérifiés pour une information reçue.

De même, certains traitement qui nécessitent beaucoup de matériel lorsqu'ils sont traités en combinatoire (conversion décimale ⇄ binaire, multiplication, division, etc...) peuvent être réalisés en mode série, élément binaire par élément binaire. On échange alors du matériel contre du temps de traitement. Il y a moins de matériel, mais le temps de traitement est plus important.

Annexe

REGISTRES A DÉCALAGES PRINCIPAUX TYPES DE CIRCUITS INTÉGRÉS

1. TABLEAU SYNTHÉTIQUE

La variété des registres à décalage intégrés est aussi grande que celle des registres de mémorisation ou des registres compteurs.

Nous avons classé dans le tableau de la figure 13.15 les circuits de la famille TTL LS en fonction de quelques critères principaux: nombre de broches du boîtier et nombre d'étages du registre, caractéristiques de commutation des étages, mode de fonctionnement, mise à zéro et sorties trois états. Nous avons ainsi dégagé cinq familles caractérisées par leurs modes de fonctionnement.

Type de boîtier	Nombre de broches	Nombre d'étages	Caractéristiques de commutation			Mode de fonctionnement					Mise à zéro (clear)	Sorties 3 états	Observations
						Entrées		Sorties					
			Bascule	Horloge	Fréquence MHz	Parallèle	Série	Parallèle	Série	Bidirectionnel			
74 LS 91	20	8	RST	↑	25		S		S				entrée série sortie série
74 LS 165	16	8	RST	↑	35	A			S				entrées parallèles sortie série
74 LS 166	16	8	RST	↑	35	S			S		A		
74 LS 164	14	8	RST	↑	25		S	S			A		entrée série sorties parallèles
74 LS 395 A	16	4	RST	↓	25	S	S	S	S		A	X	universels
74 LS 195 A	16	4	RST	↑	30	S	S	S	S		A		
74 LS 295 B	14	4	RST	↓	25	S	S	S	S			X	
74 LS 96	16	5	RST	↑	10	S	S	S	S		A		
74 LS 95 B	14	4	RST	↓	36	S	S	S	S	X	A		
74 LS 194 A	16	4	RST	↑	36	S	S	S	S	X	A		universels bidirectionnels
74 LS 299	20	8	D	↑	35	S	S	S	S	X	A	X	
74 LS 325	20	8	D	↑	35	S	S	S	S	X	S	X	

RST: variante de bascule RS synchrone
S : commande synchrone
A : commande asynchrone

Fig. 13.15. — *Les registres à décalage intégrés.*

2. REGISTRE A ENTRÉE ET SORTIE SÉRIE (74 LS 91)

C'est le plus simple des registres à décalage (fig. 13.16). Son entrée série est constituée par une porte ET qui permet le calcul d'une fonction d'entrée.

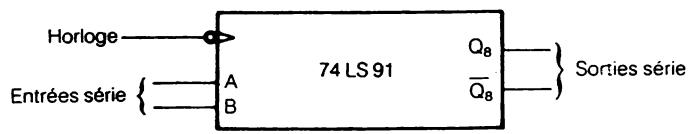


Fig. 13.16. — *Schéma bloc du registre à décalage 74 LS 91.*

3. REGISTRES A ENTRÉES PARALLÈLES ET SORTIE SÉRIE (74 LS 165 – 74 LS 166)

Ils diffèrent par le mode d'affichage (asynchrone pour le 74 LS 165, synchrone pour le 74 LS 166) et la présence d'une mise à zéro pour le 74 LS 166 (fig. 13.17).

Ce registre dispose d'une entrée d'inhibition d'horloge (*IN*) et d'une commande affichage/décalage (*SL*) sous la dépendance de l'horloge autorisant suivant son état, soit l'affichage parallèle de données dans le registre, soit le décalage des données dans le registre.

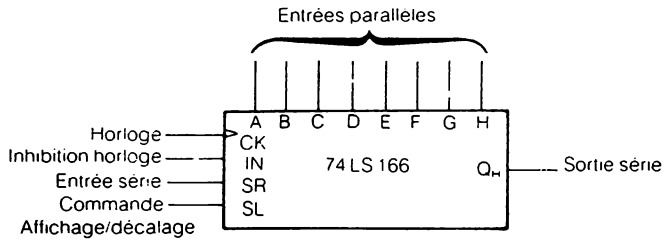


Fig. 13.17. – *Schéma bloc du registre à décalage 74 LS 166.*

4. REGISTRE A ENTRÉE SÉRIE ET SORTIES PARALLÈLES (74 LS 164)

De structure simple (fig. 13.18), il dispose sur l'entrée série d'une porte ET comme le 74 LS 91 et d'une entrée de mise à zéro asynchrone.

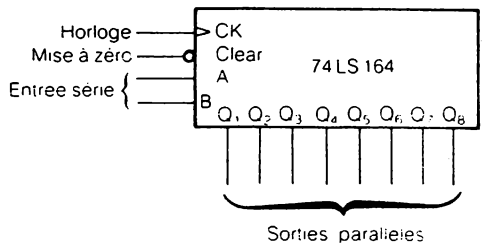


Fig. 13.18. – *Schéma bloc du registre à décalage 74 LS 164.*

5. REGISTRES UNIVERSELS – ENTRÉE ET SORTIE SÉRIE – ENTRÉES ET SORTIES PARALLÈLES
(74 LS 395 A – 74 LS 195 A – 74 LS 95 B – 74 LS 295 B – 74 LS 96)

Cette famille relativement importante permet d'effectuer tous les types de décalages ; cependant, pour un nombre de broches égal ou voisin de celui des registres précédents, le nombre d'étages des registres est généralement divisé par deux. Les variantes de registres se caractérisent par l'existence d'une commande de mise à zéro ou de sortie trois états.

Nous donnons à titre d'exemple le brochage du registre 74 LS 395 A à la figure 13.19. Comme pour le 74 LS 166, la commande affichage/décalage est synchrone de l'horloge. Les sorties parallèles étant de type trois états, le dernier étage est sorti directement en Q_D, pour la mise en cascade de boîtiers.

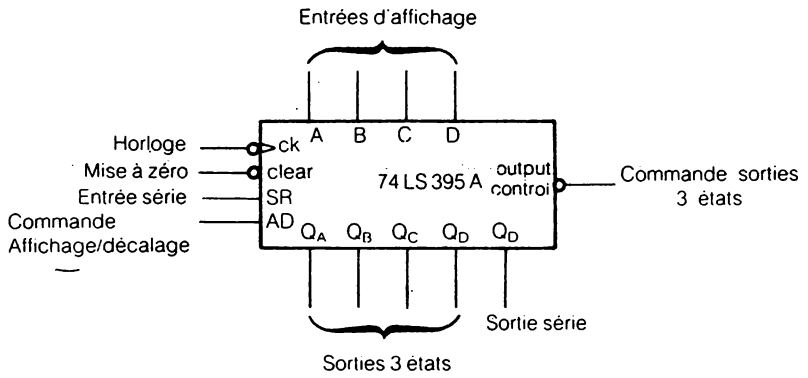


Fig. 13.19. – *Schéma bloc du registre à décalage 74 LS 395 A.*

6. REGISTRES UNIVERSELS BIDIRECTIONNELS
(74 LS 194 A – 74 LS 299 – 74 LS 323)

Registres plus complexes, leurs commandes de décalage, chargement et inhibition sont sous le contrôle d'entrées programmables (S_1 S_0). Les registres 74 LS 299 et 74 LS 323 disposent en outre d'entrées-sorties 3 états qui permettent de loger un registre universel à 8 eb dans un boîtier à 20 broches.

La figure13.20 donne le schéma de brochage du 74 LS 194 A et la figure 13.21, le schéma commun au 74 LS 299 et au 74 LS 323 qui diffère du précédent par une commande de mise à zéro synchrone. Rappelons que le fonctionnement du 74 LS 299 est décrit plus en détail au paragraphe 6.2 de ce chapitre.

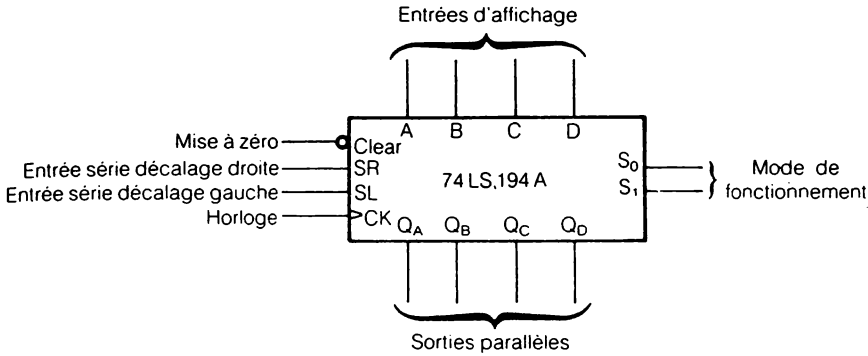


Fig. 13.20. — *Schéma de brochage du registre à décalage 74 LS 194 A.*

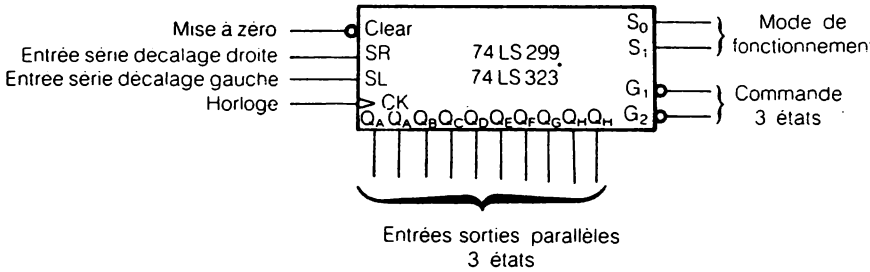


Fig. 13.21. — *Schéma de brochage des registres à décalage 74 LS 299 et 74 LS 323.*

Chapitre 14

ASSEMBLAGES SÉQUENTIELS CALCULS TEMPORELS

1. INTRODUCTION

Dans les chapitres 10, 11, 12, 13 ont été présenté les fonctions séquentielles simples (bistable, bascule, monostable, oscillateur) puis des fonctions plus complexes : mémorisation, comptage, décalage. Nous examinons dans ce chapitre des montages séquentiels fréquemment utilisés, en étudiant pour certains d'entre eux l'influence des paramètres temporels (temps de préaffichage, de maintien, etc...) sur le fonctionnement ou les performances. La prise en compte des paramètres temporels pour un montage est en effet aussi importante que les aspects logiques et électriques.

2. LE BISTABLE ANTI-REBOND

Lorsque l'on introduit une information binaire dans une machine au moyen de boutons ou d'interrupteurs, il y a généralement des rebondissements au moment où s'établissent et se rompent les contacts mécaniques. La figure 14.1 montre une manœuvre d'un interrupteur entre deux contacts *A* et *B*.

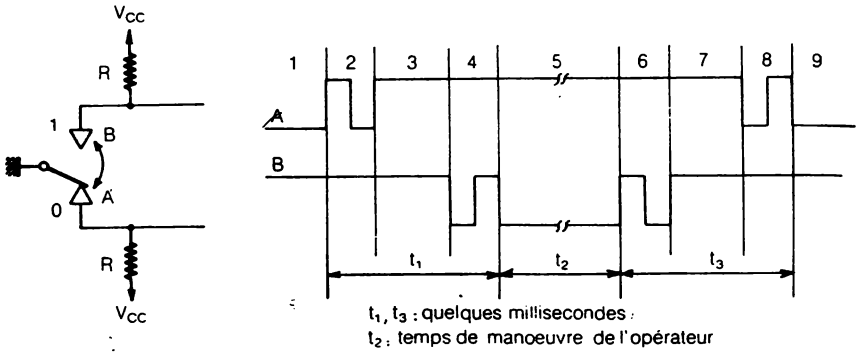


Fig. 14.1.

situation 1 : la lame de contact est sur A,
situation 2 : la lame se décolle et rebondit sur A,
situation 3 : la lame se trouve entre A et B,
situation 4 : la lame arrive sur B et rebondit,
situation 5 : la lame se trouve sur B,
situation 6 : la lame se décolle de B et rebondit,
situation 7 : la lame est entre A et B,
situation 8 : la lame arrive sur A et rebondit,
situation 9 : la lame se trouve sur A.

Si l'information fournie sur A ou B est utilisée en dehors des périodes transitoires, les rebondissements n'ont aucun effet nuisible. Par contre si le signal sert à activer une entrée dynamique de circuit séquentiel (horloge, remise à zéro, etc...) les rebondissements deviennent gênants : par exemple un compteur avancera de plus d'un pas si on l'incrémente en utilisant le bouton poussoir de la figure 14.1. Il faut alors filtrer le signal. La figure 14.2 donne le schéma d'un montage dit anti-rebond qui élimine les signaux transitoires.

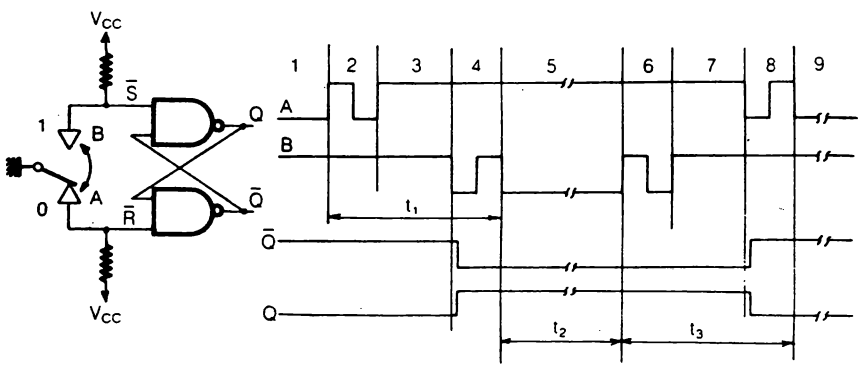


Fig. 14.2. — *Montage anti-rebond.*

A l'origine (situation 1) la lame est sur A, le bistable se trouve en 0.

Dans la situation 2, la lame décolle de A mais y revient une ou plusieurs fois. L'entrée $\overline{R}$ reçoit donc une succession de 0 et 1. Les 1 n'agissent pas et les 0 confirment l'état 0 du bistable.

Dans la situation 3, la lame se trouve entre A et B. Les entrées $\overline{R}$ et $\overline{S}$ du bistable sont polarisées avec deux 1, ce qui n'influence pas le bistable qui conserve son état 0.

A la situation 4, la lame arrive sur B et rebondit. L'entrée $\overline{S}$ reçoit une succession de 1 et de 0. Les 1 n'agissent pas, mais le premier 0 fait passer le bistable à 1 et les autres restent ensuite sans effet.

A la situation 5, la lame se trouve sur B de manière stable : $\overline{S}$ vaut 0, $\overline{R}$ vaut 1, le bistable reste à 1.

A la situation 6, la lame décolle de B et rebondit. La succession de 1 et 0 sur $\overline{S}$ est sans effet sur le bistable qui reste à 1.

A la situation 7, la lame se trouve entre A et B, $\overline{R} = \overline{S} = 1$ ce qui laisse le bistable en 1.

A la situation 8, la lame arrive et rebondit sur A. L'entrée $\overline{R}$ reçoit une succession de 1 et de 0. Le premier 0 met le bistable à 0.

A la situation 9 enfin, l'entrée $\overline{R}$ vaut 0 et $\overline{S}$ vaut 1 de manière stable. Le bistable demeure dans la position 0.

On voit à la figure 14.2 que les sorties Q et $\overline{Q}$ du bistable reproduisent les états stables de A et B, sans présenter leurs états transitoires. On note aussi que ce montage n'élimine que les rebondissements au voisinage d'un contact, mais pas les rebondissements d'un contact à un autre. Ce dernier cas correspond à un bouton mécaniquement défectueux.

3. AFFICHAGE EN 1 OU 0 D'UN BISTABLE OU D'UNE BASCULE

Supposons que l'on veuille mettre en 1 ou en 0 un élément de mémoire B, avec deux conditions M et R en coïncidence avec une horloge h, selon les équations :

$M.h \rightarrow B$ à 1

$R.h \rightarrow B$ à 0

On suppose que M et R ne valent jamais 1 en même temps et sont stables autour de l'impulsion d'horloge. La figure 14.3 montre le diagramme désiré.

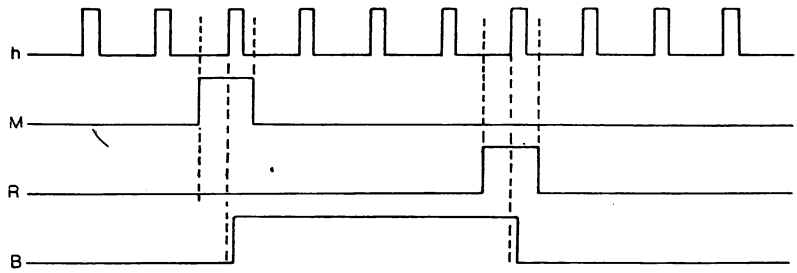


Fig. 14.3. — *Mise en 1 et 0 d'un point mémoire.*

3.1. Bistable RS

Avec un bistable RS, le schéma est évident : on applique $M.h$ sur S et $R.h$ sur R (fig. 14.4).

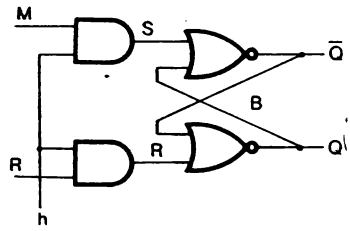


Fig. 14.4 — *Mise en 1 et 0 d'un bistable.*

3.2. Bascule JK

Avec une bascule JK, on peut utiliser les entrées asynchrones (fig. 14.5a) ou synchrones (fig. 14.5b).

Pour les entrées asynchrones on applique $\overline{M.h}$ sur Preset et $\overline{R.h}$ sur Clear. Pour les entrées synchrones, on applique respectivement M, R, h sur J, K et horloge. Quand $M = R = 0$, $J = K = 0$, la bascule conserve son état. Quand $M = 1$, $R = 0$, $J = 1$, $K = 0$, la bascule passe à 1. Quand $M = 0$, $R = 1$, $J = 0$, $K = 1$, la bascule passe à 0. Le cas $M = R = 1$ qui ferait

commuter la bascule ne se rencontre pas par hypothèse (voir fig. 14.3).

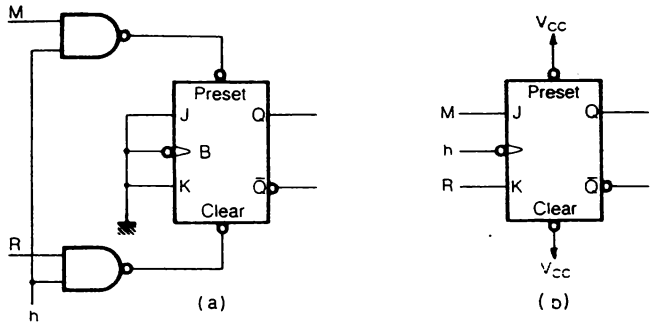


Fig. 14.5. — *Mise en 0 ou 1 d'une bascule JK.*

3.3. Bascule D

S'il s'agit d'une bascule D, on peut comme pour la bascule JK se servir des entrées asynchrones (fig. 14.6).

$\overline{R.h}$ sur entrée Clear (inverse).

$\overline{M.h}$ sur entrée Preset (inverse).

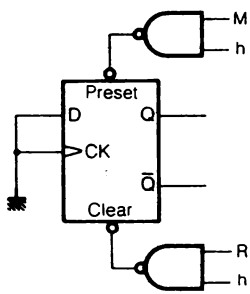


Fig. 14.6. — *Mise en 1 et 0 d'une bascule D.*

Par contre l'utilisation de D et clock n'apparaît pas évidente. Une première solution est d'élaborer une horloge aux instants de mise en 1 et mise à 0, et d'appliquer un état 1 sur l'entrée D seulement quand on veut mettre la bascule en 1. Cela conduit à appliquer :

M sur l'entrée D,

$(M + R).h$ sur l'entrée horloge (fig. 14.7).

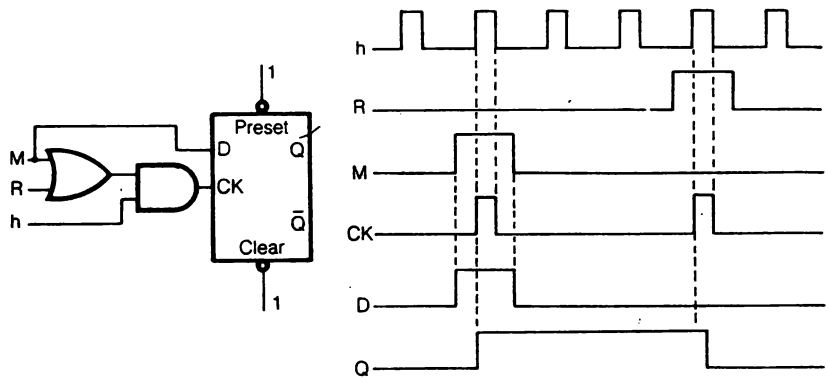


Fig. 14.7. — *Mise à 1 et 0 synchrone d'une bascule D avec horloge conditionnelle.*

Une autre solution utilisant les entrées synchrones consiste à appliquer une horloge permanente sur l'entrée CK et à placer sur D une fonction, dépendant de R, M et Q qui assure le maintien à 1 après la disparition du signal M. Établissons un tableau de vérité donnant Q^+ en fonction de R, M et Q (fig. 14.8).

R	M	Q	Q^+	D	Situation
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	2
0	1	1	1	1	3
1	0	0	0	0	4
1	0	1	0	0	5
1	1	0	X	X	6
1	1	1	X	X	7

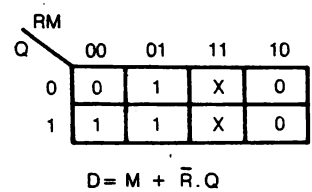


Fig. 14.8. — *Mise en 1 et 0 synchrone d'une bascule D avec une horloge permanente: Table de vérité.*

Situations 0 et 1 : $R = M = 0$. On maintient l'état ancien de Q.

Situations 2 et 3 : $R = 0$, $M = 1$. On met $Q^+ = 1$.

Situations 4 et 5 : $R = 1$, $M = 0$. On met $Q^+ = 0$.

Siguations 6 et 7 : $R = 1$, $M = 1$. Impossible par hypothèse.

On obtient $Q^+ = D = M + \bar{R}Q$. La figure 14.9 donne le schéma et le diagramme correspondant.

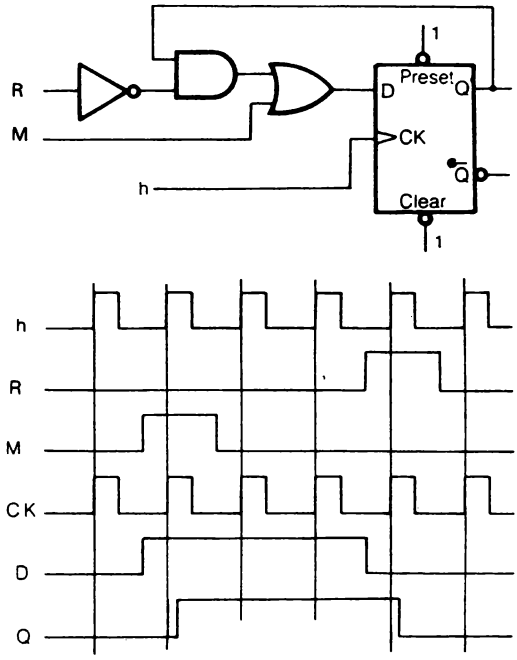


Fig. 14.9 — *Mise en 1 et 0 synchrone d'une bascule D avec horloge permanente. Schéma et diagramme.*

En conclusion la bascule D n'apparaît pas bien adaptée à la commande par deux signaux R et M de manière synchrone. Il faut lui préférer la bascule JK.

4. TRANSFERT D'UNE INFORMATION DANS UN POINT MÉMOIRE

Soit une information I à transférer dans un point mémoire B avec horloge h. Cela s'écrit :

$h \rightarrow I$ dans B

Avec un bistable RS cette équation devient :

$h.I \rightarrow Q$ à 1

$h.\overline{I} \rightarrow Q$ à 0

d'où le schéma de la figure 14.10a, qui correspond d'ailleurs à un bistable D.

Avec une bascule JK nous pouvons écrire :

$$Q^+ = J\bar{Q} + \bar{K}Q$$

$$Q^+ = I = I(\bar{Q} + Q) = I\bar{Q} + IQ$$

d'où $J = I$ et $K = \bar{I}$ (fig. 14.10b).

Avec une bascule D nous pouvons écrire :

$$Q^+ = D$$

$Q^+ = I$ d'où $D = I$ (fig. 14.10c).

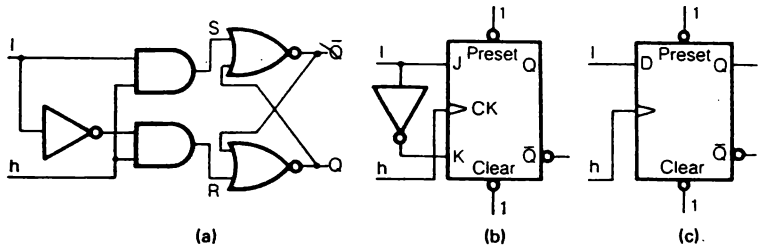


Fig. 14.10. — *Transfert d'information dans un point mémorisé.*

Pour cette fonction, la bascule D est la mieux adaptée.

5. BISTABLES A ENTRÉES MULTIPLES

Il s'agit de déterminer les fonctions assurées par les montages de la figure 14.11.

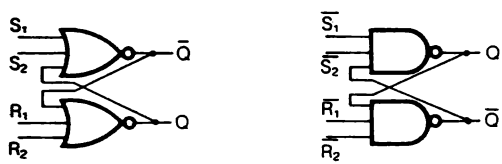


Fig. 14.11. — *Bistables à entrées multiples.*

D'après le schéma il vient pour le bistable réalisé avec des portes NOR.

$$\bar{Q}^+ = \overline{Q + S_1 + S_2} = \overline{Q + (S_1 + S_2)} = \overline{Q + S}$$

$$Q = \overline{\bar{Q} + R_1 + R_2} = \overline{\bar{Q} + (R_1 + R_2)} = \overline{\bar{Q} + R}$$

donc $S = S_1 + S_2$, $R = R_1 + R_2$

Pour le bistable réalisé avec des portes NAND, il vient :

$$Q^+ = \overline{\overline{S_1} \cdot \overline{S_2} \cdot \overline{Q}} = \overline{(\overline{S_1} + \overline{S_2}) \cdot \overline{Q}} = \overline{\overline{S} \cdot \overline{Q}}$$

$$\overline{Q}^+ = \overline{\overline{R_1} \cdot \overline{R_2} \cdot Q} = \overline{(\overline{R_1} + \overline{R_2}) \cdot Q} = \overline{\overline{R} \cdot Q}$$

donc $S = S_1 + S_2$ et $R = R_1 + R_2$

Dans les deux cas les bistables de la figure 14.11 sont équivalents à des bistables simples précédés d'une porte OU réunissant les entrées de commande (fig. 14.12).

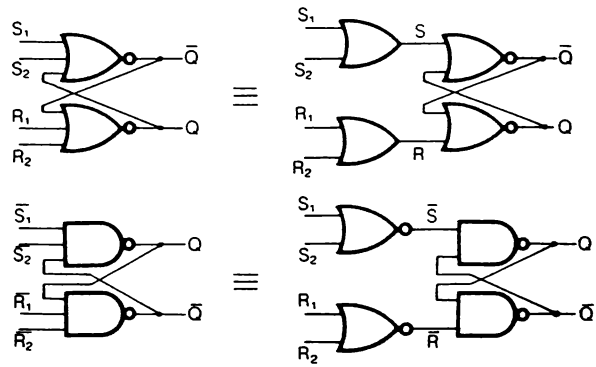


Fig. 14.12. — *Équivalence entre bistables.*

6. PARAMÈTRES TEMPORELS DES BASCULES

6.1. Rappel des paramètres temporels

Nous examinons dans ce paragraphe l'incidence des paramètres temporels sur les montages. Les paramètres de fonctionnement d'une bascule sont :

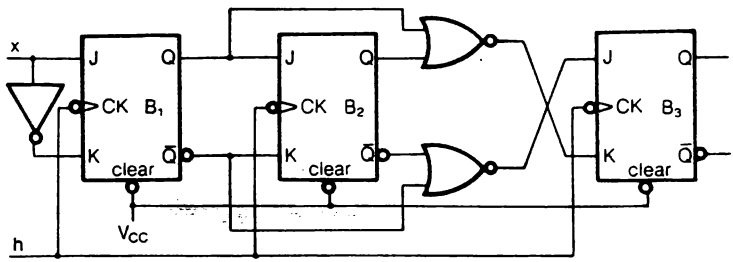
- la fréquence maximale de l'horloge : f_{ck},
- les durées minimales des impulsions d'horloge, de mise en 1 et de mise à 0 : t_w,
- le temps de préaffichage des entrées synchrones par rapport à l'horloge : t_{su},

— le temps de maintien des entrées synchrones par rapport à l'horloge : t_h,

— les temps de propagation d'une entrée vers une sortie : t_{PLH} et t_{PHL}.

6.2. Calcul de la fréquence maximale de fonctionnement

Le constructeur donne la fréquence maximale de fonctionnement d'une bascule seule, sans connexion avec l'environnement. En fait, un montage comprenant des bascules et des portes fonctionne bien en-deçà de la fréquence maximale comme nous allons le voir sur un montage, à cause des contraintes imposées par tous les circuits. Calculons la période minimale de l'horloge du montage de la figure 14.13, comprenant trois bascules B_1, B_2, B_3.



B1,B2,B3 : bascules 74 LS 73 A
portes 74 LS 02

Fig. 14.13. — *Exemple de montage.*

La figure 14.14 indique le comportement des différents éléments. Après un front de descente du signal d'horloge, les bascules B_1 et B_2 commutent avec un retard *tp*, situé entre une valeur minimale et maximale. Entre ces deux valeurs l'état de la bascule n'est pas défini. Les portes NOR calculent les fonctions B_1 B_2 et $\overline{B}_1$ $\overline{B}_2$ appliquées aux entrées J et K de B_3. Ces fonctions sont retardées par les portes et ne sont pas définies entre :

$t_{p\ min\ (Bascule)} + t_{p\ min\ (Porte)}$ et les valeurs $t_{p\ max\ (Bascule)} + t_{p\ max\ (Porte)}$.

Les fonctions présentées sur les entrées J et K de B_3 doivent être stables un temps t_{su} avant le front descendant d'horloge et un temps t_h après ce même front. Le temps de maintien t_h est automatiquement respecté car il est nul. La période T séparant deux fronts s'écrit :

$$T = t_{p\ (B)} + t_{p\ (P)} + t_{su\ (B)} + \text{marge.}$$

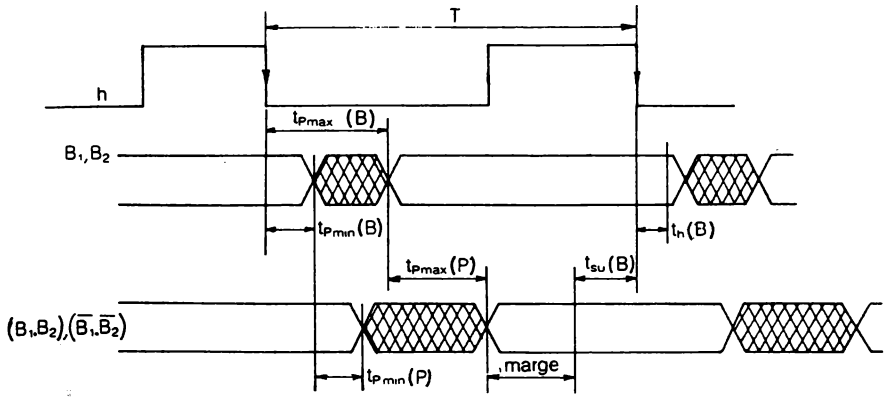


Fig. 14.14.

La marge sert à tenir compte d'une dispersion des paramètres temporels qui sont spécifiés à 25° C, 5 *V*, et pour une charge résistive et capacitive donnée (15 *pF*, 2 *kΩ* pour les éléments choisis) alors que la charge du montage peut être différente. Le pire des cas (T_{min}) s'obtient en prenant $t_{p\,(B)}$, $t_{p\,(P)}$, t_{su} à leur valeur maximale.

$$T_{min} = t_{p\ max\ (B)} + t_{p\ max\ (P)} + t_{su\ max\ (B)} + \text{marge}$$

Avec les circuits choisis on obtient :

$$T_{min} = 20 + 15 + 20 + \text{marge} = 55 + \text{marge.}$$

Prenons une marge de 50 % il vient :

$$T_{min} = 55 + 27{,}5 = 82{,}5\ ns$$

d'où $F_{max} = 12{,}12\ MHz$ ou 18,18 *MHz* sans prendre de marge alors que la fréquence maximale de la bascule 74 LS 73 A est de 30 *MHz*.

La fréquence de fonctionnement d'un montage est d'autant plus basse qu'il y a des circuits combinatoires en cascade entre deux circuits séquentiels.

6.3. Temps de préaffichage et de maintien

Une condition de bon fonctionnement d'une bascule est de conserver les données stables un temps $t_{set\ up}$ avant le front actif d'horloge et un temps

t_{hold} après le même front. Le montage de la figure 14.15 ne respecte pas ces conditions. Ce montage enregistre l'information x dans la bascule B_1 à chaque impulsion d'horloge et transfère le contenu B_1 dans B_2 avec la même horloge h si la condition C vaut 1.

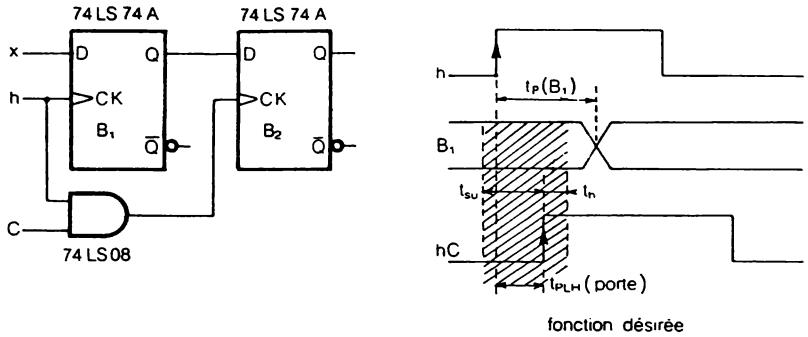


Fig. 14.15. — *Montage ne respectant pas les paramètres t_{su} et t_h.*

L'horloge hC de B_2 est décalée par rapport à h du retard de la porte ET calculant hC. L'information entrant dans B_2 est décalée par rapport à h du retard de la bascule B_1.

Si la bascule B_1 est rapide et la porte ET lente, l'information entrante ne respectera pas le temps de préaffichage (fig. 14.16a). Si la bascule B_1 a un temps de réponse voisin de celui de la porte ET, le temps de maintien ne sera pas respecté (fig. 14.16b).

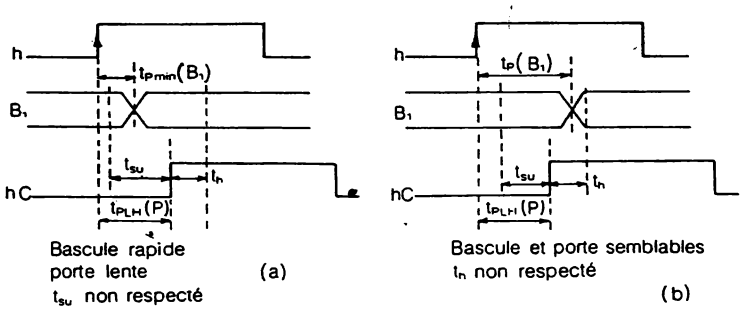


Fig. 14.16. — *Non respect des t_{su} et t_h.*

Les valeurs des paramètres des bascules et portes rassemblées dans le tableau de la figure 14.17 montrent que les configurations de la figure 14.16 sont possibles.

composants	Paramètres (ns)					
	t_{PLH}		t_{PHL}		t_{su}	t_h
	TYP	MAX	TYP	MAX		
bascule 74 LS 74 A	13	25	25	40	20 ou 25	5
porte ET 74 LS 08	8	15	10	20	—	—

Fig. 14.17. — *Valeurs de paramètres temporels.*

Le montage de la figure 14.15 présente donc un aléa. La figure 14.18 propose une solution avec la même horloge appliquée aux deux bascules. Lorsque $C = 0$, on valide là recopie de B_2 dans elle-même au front d'horloge h. Si $C = 1$, c'est l'état B_1 qui est copié dans B_2 au front d'horloge de h. Comme la même horloge est appliquée sur B_1 et B_2 il n'y a plus les problèmes de décalage entre la commutation de B_1 et l'horloge h.

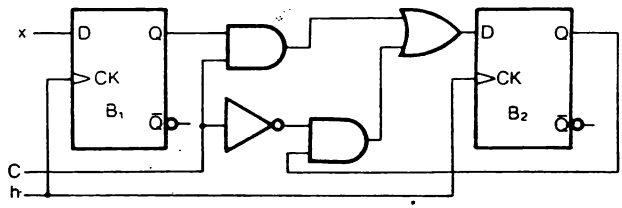


Fig. 14.18. — *Suppression d'un aléa dû au décalage d'horloge.*

6.4. Durée des signaux

Les signaux appliqués à un circuit ne doivent pas descendre en-dessous d'une certaine durée, sinon il y a risque de mauvais fonctionnement. Certains montages produisent par principe des impulsions de courte durée. Il faut les éviter. La figure 14.19 montre un de ces montages déconseillés. Quand l'horloge appliquée à B_1 passe de 1 à 0, la bascule B_1 passe à 1. La sortie $\bar{Q}$ passe à 0 et agit sur le clear B_1 de sorte qu'au bout de très peu de temps (t_{PHL} de B_1) B_1 revient à 0. On a ainsi obtenu une impulsion qui est appliquée à l'entrée horloge de B_2.

Pour une bascule B de type 74 LS 73 A, la durée minimum garantie de l'impulsion sur clear est de 25 *ns*, le retard de 15 *ns* en typique ou 20 *ns* au maximum, tandis que pour la bascule 74 LS 74 A une durée minimum de 25 *ns* est nécessaire sur l'entrée horloge. Si la bascule *JK* est un peu plus

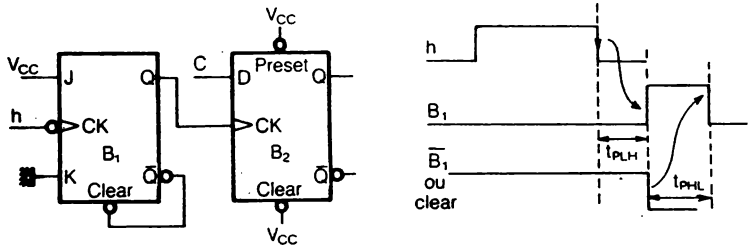


Fig. 14.19. — *Montage donnant une impulsion trop courte.*

rapide que les valeurs minimales garanties, l'impulsion sur Q de B_1 peut devenir insuffisante pour commander l'horloge de B_2 et provoquer un aléa.

7. FRÉQUENCE MAXIMALE DE FONCTIONNEMENT D'UN COMPTEUR

Quand on assemble des boîtiers de compteur en cascade, la fréquence maximale de fonctionnement décroit. La figure 14.20 montre un schéma de compteur à seize étages réalisé avec des boîtiers 74 LS 163 A. La sortie de retenue d'un boîtier est reliée aux entrées de validation EP et ET du boîtier suivant.

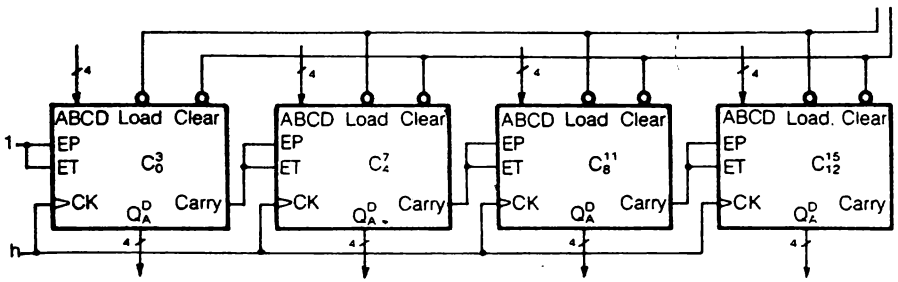


Fig. 14.20. — *Mise en cascade de boîtiers compteurs 74 LS 163 A.*

Si l'on compte le temps à partir de l'horloge on a :

- un temps t_p entre l'horloge et la retenue du premier boîtier (35 *ns* maximum),
- un temps t_p entre la validation ET et la retenue du deuxième boîtier (14 *ns* maximum) ainsi que pour le troisième boîtier,

– un temps t_{su} à respecter pour les entrées EP, ET du dernier boîtier (20 *ns*).

D'où une durée de la période de comptage :

$t_{min} = t_{P(h,r)} + 2\ t_{P(ET,r)} + t_{su}$ + marge

$t_{min} = 35 + 28 + 20 +$ marge $= 83 +$ marge.

Avec une marge de 50 % :

$t_{min} = 125\ ns$

d'où $f_{max} = 8\ MHz$. La fréquence maximale de fonctionnement d'un boîtier seul est de 25 *MHz*.

8. PRÉLÈVEMENT D'INFORMATION DANS UN REGISTRE

Supposons qu'un compteur ou un registre à décalage fonctionne avec une horloge h_1 et que des fonctions combinatoires soient établies à partir des sorties de registre. A quel moment t une horloge h_2 peut-elle exploiter ces fonctions dans un circuit séquentiel (fig. 14.21) ? Il faut attendre que la sortie du registre soit stabilisée ($t_{PR\ max}$), que le résultat du calcul combinatoire le soit aussi ($t_{PC\ max}$) et que le temps de préaffichage t_{su} du circuit séquentiel soit satisfait.

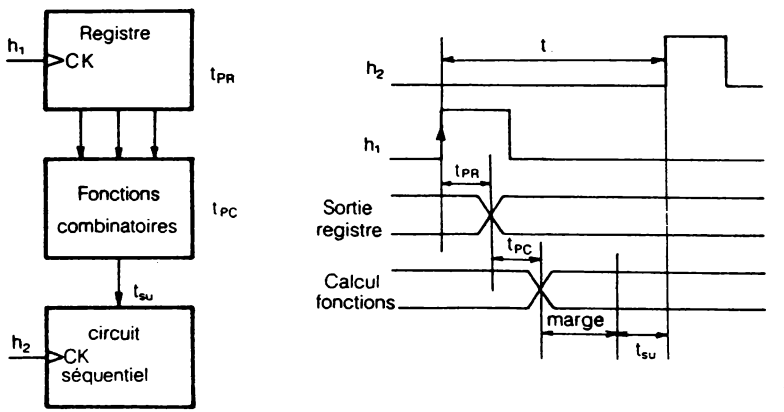


Fig. 14.21. – *Exploitation des états d'un registre.*

L'intervalle t séparant h_1 et h_2 est :

$t = t_{PR} + t_{PC} + t_{su}$ + marge.

9. RÉALISATION DE CYCLES D'HORLOGE

La figure 14.22 présente un montage donnant un cycle fixe à huit temps décalés.

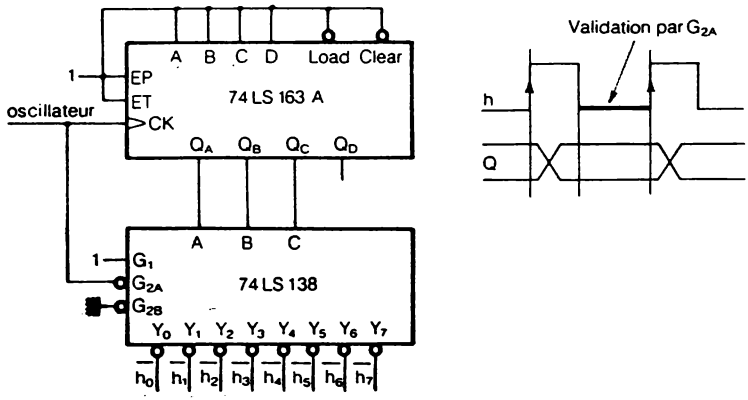


Fig. 14.22. — ***Réalisation d'un cycle d'horloge à huit temps.***

Les entrées EP, ET sont placées à 1 permanent (le compteur compte en permanence) ainsi que Load et Clear (entrées inverses, inactives avec 1). La valeur 15 présente sur $ABCD$ n'est pas utilisée puisque Load = 1.

Le décodeur reçoit les sorties QA, QB et QC, les décode après validation par l'oscillateur inversé. On obtient une série de huit temps inverses décalés, synchrones avec l'horloge inverse.

Le montage de la figure 14.23 montre comment obtenir un cycle comportant entre un et huit temps. Par rapport au montage de la figure 14.22 on a ajouté un décodeur qui détermine la fin du cycle. Le lecteur pourra s'exercer à trouver un montage donnant un cycle programmable en utilisant la sortie retenue et les entrées de programmation (A, B, C, D et Load) et calculer les fréquences maximales de fonctionnement.

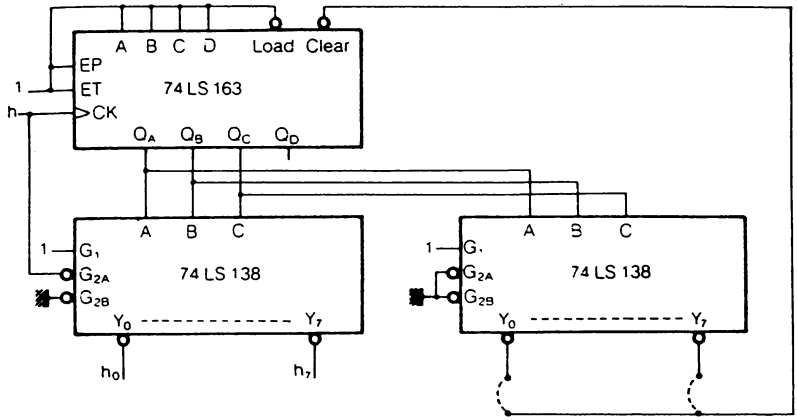


Fig. 14.23. — *Cycle d'horloge programmable.*

10. RÉALISATION D'UNE TEMPORISATION

Le principe de réalisation d'une temporisation par comptage a été donné au chapitre 12. La figure 14.24 donne un schéma avec des compteurs 74 LS 163 A. A partir d'une horloge de période 1 *ms*, on élabore une impulsion toutes les 100 *ms*.

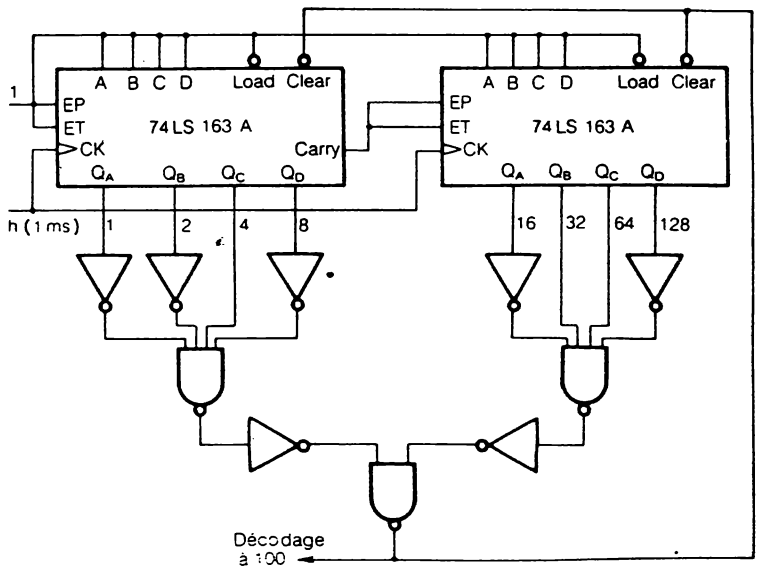


Fig. 14.24. — *Temporisation de 100 ms.*

On remarque que le décodage est plus volumineux (2,25 boîtiers) que le comptage. Une autre solution consiste à se servir du décodage à 15 des deux compteurs. Quand la sortie carry du deuxième compteur vaut 1, c'est que le contenu du premier et deuxième compteur vaut 15, soit 255 pour l'ensemble. Si n est le nombre de périodes à compter, on charge $255 - n$ dans le compteur et l'on compte n coups. On obtient $(255 - n) + n = 255$, valeur dont le décodage est disponible sur la sortie retenue des poids forts. La figure 14.25 donne le schéma correspondant.

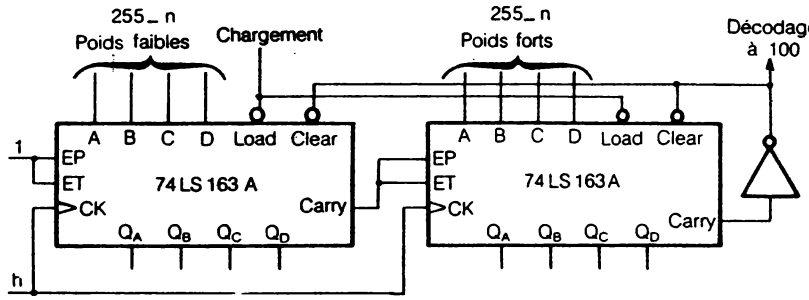


Fig. 14.25. — *Temporisation de 100 ms.*

Une autre solution voisine de celle de la figure 14.25 consiste à utiliser un décompteur dans lequel on charge la valeur de la temporisation et que l'on décompte ensuite jusqu'à 0. La retenue au moment du passage à 0 donne le signal de temporisation.

11. RÉALISATION D'UN MONOSTABLE LOGIQUE PROGRAMMABLE

Un système logique peut exploiter des mémoires (chap. 15) de différentes vitesses. Quand il formule une demande de lecture il doit attendre un certain temps pour que la mémoire puisse accéder à l'information demandée. Pour prévenir le système logique que l'information dont on a demandé la lecture est prête, on déclenche à partir de la demande de lecture une temporisation. Si l'on veut que le même montage puisse servir à des mémoires de temps d'accès variables, on peut réaliser le schéma de la figure 14.26 conçu avec des registres à décalage 74 LS 194 A.

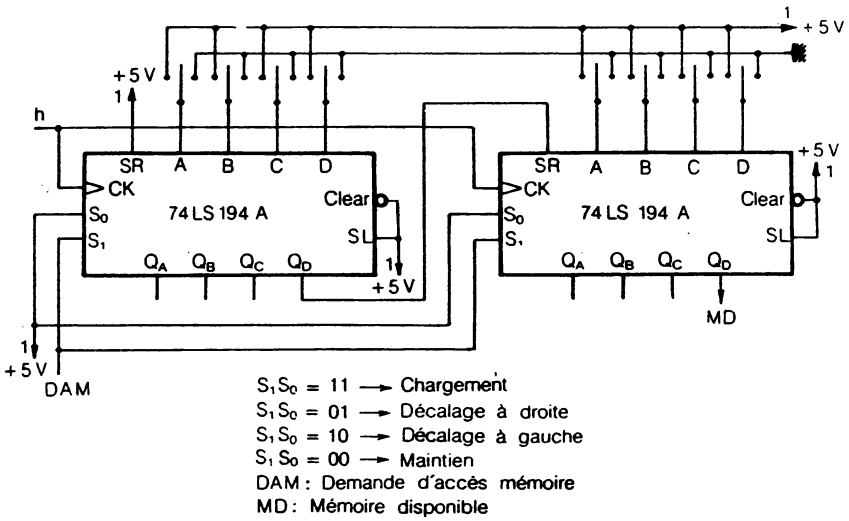


Fig. 14.26. — *Temporisation avec registre à décalage (montage).*

Au repos, il n'y a pas de demande d'accès mémoire et DAM = 0 de sorte que l'on applique S_1 $S_0 = 01$ comme commande au registre. Cela donne un décalage à droite avec un 1 entrant. Le registre est plein de 1 et le dernier étage indique par MD = 1 que la mémoire est disponible.

Survient une demande d'accès mémoire, DAM = 1. Le registre reçoit S_1 $S_0 = 11$ ce qui correspond au chargement. On charge alors une configu-

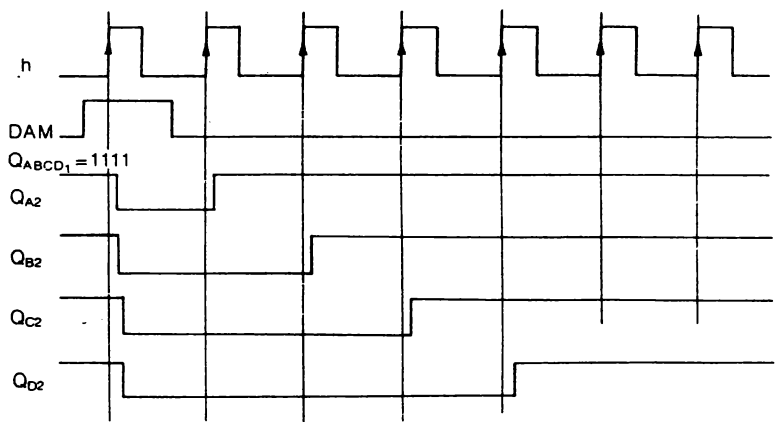


Fig. 14.27. — *Temporisation avec registre à décalage (diagramme).*

ration 11...100...0 dans le registre (1 à gauche, 0 à droite). *MD* passe ainsi à 0 (mémoire non disponible). Ensuite DAM disparaît et la fonction décalage à droite recommence. On pousse des 1 dans le registre. Quand les 1 atteignent Q_D du deuxième registre le circuit indique que la mémoire est disponible. La configuration aux entrées des registres est programmée par des mini-interrupteurs (elle pourrait également être programmé par le contenu d'un mot d'une mémoire ou d'un registre).

La durée d'indisponibilité de la mémoire est comprise entre 0 cycle d'horloge à huit cycles d'horloges (fig. 14.27).

Chapitre 15

LES MÉMOIRES PROPRIÉTÉS GÉNÉRALES

1. CAPACITÉ ET ACCESSIBILITÉ

Une bascule mémorise une information dont l'inscription s'effectue soit en mode synchrone (horloge et D, horloge et $J\,K$), soit en mode asynchrone (Preset, Clear) au choix pour la même bascule. L'état de l'information est constamment disponible sur une ou deux sorties (Q et $\overline{Q}$).

Un registre permet une mémorisation plus importante de quatre ou huit éléments binaires simultanément. Mais à cause du nombre limité de broches de boîtier (14, 16 ou 20), l'accessibilité se restreint. L'inscription s'opère de manière globale pour toutes les bascules avec une horloge commune, les données étant appliquées aux entrées d'information. Les commandes générales comme l'effacement s'appliquent également à toutes les bascules à la fois. L'affichage sélectif d'un eb dans le registre n'est pas possible et doit être réalisé à l'extérieur du registre (lecture, modification, réécriture). Enfin, pour réduire le nombre de broches, on peut faire jouer deux rôles à certaines d'entre elles : servir aux entrées et aux sorties. Ceci conduit à des montages à commande trois états.

Pour des capacités plus importantes (16, 32 eb...), on associe quelques boîtiers de registres. Mais on se limite pratiquement à quelques dizaines d'éléments binaires.

Pour des capacités de plusieurs centaines ou milliers d'eb, on a recours à des mémoires. Les mémoires sont des ensembles qui comprennent un grand nombre de cellules élémentaires, stockant chacun une information ; cela peut être une bascule, une capacité, une métallisation ou une absence de métallisation dans un circuit, un domaine magnétique sur un support mobile, un domaine magnétique ou une charge magnétique se déplaçant dans une structure, etc... Quelle que soit la technique de réalisation, l'accessibilité est encore moins grande qu'avec les registres. Pour écrire ou lire une information, il faut spécifier le numéro de la cellule ou définir la position physique du domaine contenant l'information, et c'est après une opération plus ou moins complexe, faisant intervenir plusieurs signaux que l'on pourra lire le contenu d'une case de la mémoire ou modifier son contenu.

2. LES GRANDES CATÉGORIES DE MÉMOIRES

De nombreuses techniques ont été mises à profit dans le passé pour réaliser des mémoires. Aujourd'hui, un tri s'est opéré, et il ne reste plus que quelques grandes catégories.

2.1. Les mémoires à semi-conducteur

Elles sont nées vers 1970, avec les progrès de l'intégration des semi-conducteurs. Elles utilisent les mêmes procédés de fabrication que ceux employés pour les circuits logiques. Ce sont celles qui se développent le plus. Comme nous le verrons plus loin, il en existe de nombreuses variétés. Les boîtiers les plus denses actuellement disponibles (1985) mémorisent 256 000 eb. Ces mémoires à fonctionnement entièrement électronique ont un temps d'accès à l'information très rapide (quelques dizaines de manosecondes pour les plus rapides).

2.2. Les mémoires magnétiques

Ces mémoires sont plus anciennes que les précédentes et remontent au début des années 1960. De forme géométrique variée, (tambour, bande, disque), elles stockent les éléments binaires sous la forme d'un domaine

aimanté et c'est par déplacement mécanique du support devant des têtes de lecture-écriture que l'on a accès à l'information. Les capacités sont élevées, 400 millions d'octets (1 octet = 8 eb) pour un disque magnétique par exemple, mais les temps d'accès à l'information sont longs à cause des mouvements mécaniques: plusieurs dizaines de millisecondes. On les appelle aussi mémoires de masse à cause du grand volume d'information stockée.

2.3. Les mémoires à bulles magnétiques

Beaucoup plus récentes (fin des années 1970), ces mémoires utilisent le déplacement d'un domaine magnétique sous l'influence d'un champ magnétique dans une structure métallique déposée sur un support. La mémoire fonctionne comme un très grand registre à décalage dont la sortie est rebouclée sur l'entrée. Il n'y a aucun mouvement mécanique. Actuellement les constructeurs proposent des boîtiers d'un million eb et les circuits de gestion associés. Si ces mémoires se développent, elles devraient concurrencer les mémoires magnétiques (élimination des temps de déplacement mécanique) plutôt que les mémoires à semi-conducteurs (le fonctionnement en registre à décalage ne permet pas une grande vitesse d'accès à l'information).

Dans ce chapitre, nous nous intéressons seulement aux mémoires à semi-conducteur.

3. TECHNOLOGIES DES MÉMOIRES A SEMI-CONDUCTEUR

Il y a quelques grandes technologies communes à toutes les mémoires et ensuite, selon les constructeurs, une multitude de variantes liées à l'expérience de chacun d'eux.

3.1. Technologie MOS ou Metal Oxide Semi conductor

Les circuits MOS sont des transistors à effet de champ qui permettent grâce à leur taille réduite, des densités d'intégration élevées et grâce aux hautes impédances mises en jeu, des consommations faibles. En contrepartie, les hautes impédances et les inévitables capacités parasites de structure donnent des constantes de temps plus élevées, ce qui diminue la vitesse de fonctionnement.

Des améliorations successives (utilisation de MOS canal *N* au lieu de MOS canal *P*) ont permis de réduire l'écart avec les techniques bipolaires.

3.2. Technologie CMOS ou MOS Complémentaire

En associant des transistors MOS canal *P* et MOS canal *N*, on obtient des structures consommant très peu de courant. Comme la mémoire représente de plus en plus la majeure partie d'un système, diminuer sa consommation amène des économies importantes. En contrepartie, la taille des cellules est plus grande et ne permet pas les densités d'intégration des MOS ni leur vitesse.

3.3. Technologie bipolaire

Il s'agit de la même technologie que celle utilisée dans les circuits logiques. La consommation est importante, la vitesse élevée, et les densités plus faibles qu'en technologie MOS. On trouve des mémoires TTL ou ECL ou des mémoires avec structure ECL et interface TTL pour faciliter les connexions. On note toutefois que les performances des mémoires MOS ont tendance à se rapprocher de celles des bipolaires.

4. CLASSEMENT DES MÉMOIRES A SEMI-CONDUCTEUR SELON L'UTILISATION

4.1. Généralités

Il existe de nombreuses variétés de mémoires à semi-conducteur. Cette diversité tient à ce que la mémoire idéale de grande capacité, consommant peu d'énergie, de vitesse élevée, gardant son information en cas de coupure d'alimentation, n'existe pas. Les différentes catégories de mémoires que nous rencontrerons sont des compromis sur quelques paramètres indispensables pour une application alors que d'autres organisations privilégient d'autres paramètres. Dans ce chapitre, nous classons les mémoires en fonction de leur utilisation.

4.2. La mémoire vive ou RAM (Random Access Memory)

Une mémoire vive est une mémoire dans laquelle on peut écrire ou lire une information. Il s'agit du prolongement des mémoires à registre que nous évoquions plus haut. En anglais, on la désigne sous le sigle RAM (Random Access Memory), mémoire à accès aléatoire ; cela signifie qu'après avoir lu ou écrit une cellule donnée, on peut lire ou écrire une cellule dont la position par rapport à la précédente est quelconque. Ceci s'oppose aux mémoires à accès série dans lesquelles après avoir lu ou écrit une cellule, la prochaine opération ne peut porter que sur la cellule de position immédiatement voisine. Le contenu d'une mémoire vive s'efface si la tension d'alimentation disparaît, aussi les qualifie-t-on parfois de mémoires volatiles.

4.3. La mémoire morte ou ROM (Read Only Memory)

L'utilisateur ne peut que lire cette mémoire. Elle est inscrite par le constructeur au moment de la fabrication selon les spécifications du client. Cette mémoire s'utilise lorsque les données qu'elle contient ne varient pas, comme par exemple la liste des valeurs de la fonction sinus pour les angles compris entre 0 et 90°. Si malgré tout, le contenu change, il faut commander un autre boîtier au constructeur, ce qui demande beaucoup de temps (plusieurs semaines).

L'utilisation de mémoires ROM dans une application ne devient intéressante que si le nombre de boîtiers identiques est grand (plusieurs milliers), compte tenu du coût de développement initial (masque du contenu de la mémoire).

Le gros avantage des mémoires ROM est de conserver leur contenu après une coupure d'alimentation.

4.4. La mémoire programmable par l'utilisateur ou PROM (Programmable Read Only Memory)

Lorsqu'une infomation peut varier de temps en temps, la mémoire ROM ne convient pas. On utilise alors des mémoires PROM, programmables par l'utilisateur. au moyen d'un dispositif adéquat appelé programmeur. Si après inscription et utilisaion, le contenu s'avère inexact, on change

le boîtier et on en reprogramme un autre. Cela prend quelques minutes. Ce type de boîtier est intéressant pour des petites séries. Comme pour les mémoies ROM, le contenu ne s'altère pas après une coupure d'alimentation.

4.5. La mémoire reprogrammable (REPROM ou EPROM)

Avec les mémoires PROM, s'il faut changer le contenu, on doit jeter le boîtier et en programmer un autre. Cela peut devenir gênant du point de vue financier si ces modifications sont trop fréquentes. Les mémoires reprogrammables (REPROM), encore appelées ROM effaçables (EPROM : Electrically Programmable Read Only Memory) constituent une solution à ce problème. Elles sont effacées par exposition à une source d'ultra-violet, à travèrs une fenêtre de quartz placée sur le boîtier et inscrites électriquement, avec des tensions plus élevées que la tension d'alimentation normale. L'effacement dure une vingtaine de minutes. L'intégrité de l'information contenue dans ces mémoires est conservée après une coupure d'alimentation.

4.6. Les mémoires ROM effaçables électriquement (EEPROM)

Ces mémoires présentent l'avantage d'être insensibles aux coupures de tension comme les ROM ou PROM précédentes, d'être inscriptibles électriquement et effaçables électriquement (EEPROM: Electrically Erasable Programmable Read Only Memory). Cela permet d'effacer l'information inadéquate sur la carte même où se situe le boîtier alors qu'avec les EPROM, il faut extraire les boîtiers et les insoler aux rayons ultra-violets.

4.7. Mémoire FIFO ou file

L'abréviation FIFO signifie First In, First Out : premier entré, premier sorti. Une mémoire FIFO fonctionne comme une file devant un guichet : le premier arrivé est servi le premier, les autres attendent, et au fur et à mesure que les clients de tête sont servis, les autres progressent dans la file. L'ordre chronologique d'entrée est respectée en sortie. Une autre analogie de fonc-

tionnement est donnée par un silo, chargé par le haut et vidé par le bas ou encore par un distributeur de chewing-gum.

4.8. Mémoire LIFO ou pile

L'abréviation LIFO signifie Last In, First Out : dernier entré, premier sorti. Une mémoire LIFO fonctionne comme une pile d'assiettes : la dernière assiette posée sur le dessus de la pile est la première à en être retirée.

5. LE POINT MÉMOIRE

Selon les différentes catégories de mémoires, on trouve des cellules mémoires différentes :

5.1. Cellule statique d'une mémoire vive

Dans ce cas, l'information est stockée dans une bascule. Comme cette bascule conserve indéfiniment l'information qui y est stockée, tant que la tension d'alimentation existe bien sûr, on qualifie la mémoire de *statique.* Cependant, cette cellule présente l'inconvénient d'occuper beaucoup de place car il faut au moins 4 transistors par cellule.

5.2. Cellule dynamique d'une mémoire vive

L'information est stockée sous la forme d'une charge électrique dans une capacité de structure.

Un "1" logique correspond à la présence d'une charge et un "0» logique à son absence. Cependant, une telle structure, si elle occupe peu de place en surface, ne présente pas d'état stable. La charge électrique stockée a tendance à disparaître par suite des résistances de fuites associées à la capacité. Il faut donc constamment la rénégérer. Pour cela, on lit la cellule à intervalle fixe (quelques millisecondes) et on la réinscrit avant que la charge n'ait atteint un niveau trop bas. Pour cette raison, la mémoire est dite *dynamique* et l'opération de restauration de la charge s'appelle

rafraîchissement. Les mémoires dynamiques sont quatre fois plus denses que les mémoies statiques de même technologie mais plus délicates d'exploitation.

5.3. Cellule de mémoire ROM

Il s'agit essentiellement d'une métallisation, présente ou absente, permettant de lire un "1" ou un "O".

5.4. Cellule de mémoire PROM

La métallisation précédente est remplacée par un micro fusible que l'on peut laisser intact ou faire fondre.

5.5. Cellule de mémoire EPROM

La cellule est un dispositif qui stocke une charge qui se maintient isolée. Cette charge isolée se conserve indéfiniment (contrairement à une mémoire vive dynamique). Elle s'efface par exposition aux rayons ultra-violets.

5.6. Cellule de mémoire FIFO

Il s'agit d'une bascule, comme pour une mémoire vive statique.

6. ORGANISATION DES CELLULES MÉMOIRES

6.1. Structure matricielle

Pour les mémoires vives (statiques ou dynamiques), les mémoires ROM, PROM, EPROM, les cellules sont organisées en matrice XY (fig. 15.1). Une cellule est repérée par un numéro de ligne et un numéro de

colonne. Le nombre global qui définit ainsi la position de la cellule s'appelle *adresse*. Ainsi, une mémoire de 4096 eb possède une adresse globale de 12 eb A_0^{11} ($2^{12} = 4096$). Si l'on organise les 4096 points en une matrice carrée de 64 lignes de 64 points chacune ($64 \times 64 = 4096$), un point est sélectionné par une adresse ligne 1 parmi 64 et une adresse colonne 1 parmi 64. L'adresse ligne 1 parmi 64 provient du décodage des six premiers eb d'adresse A_0^5, ($2^6 = 64$) et l'adresse colonne provient du décodage des six derniers eb d'adresse A_6^{11}. Le découpage en matrice carrée n'est pas une obligation et un découpage en matrice rectangulaire peut exister aussi. Dans le cas précédent, on pourrait avoir aussi une matrice de 128 lignes de 32 points chacune. L'exploitation des mémoires vives dynamiques dépend beaucoup de cette organisation matricielle comme nous le verrons plus loin.

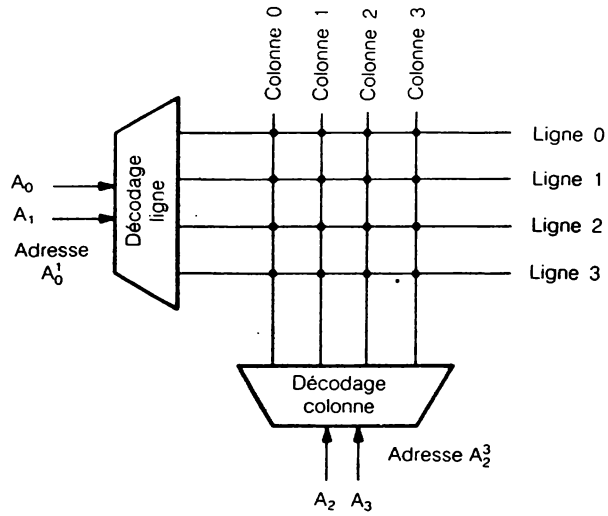


Fig. 15.1. — *Structure matricielle à* 2^4 eb.

6.2. Structure linéaire

Une mémoire FIFO (1er entré, 1er sorti) a une structure linéaire. Il s'agit d'une juxtaposition de registres à décalage (fig.15.2). L'utilisateur a accès au premier étage et au dernier. Une logique de gestion prévient l'utilisateur quand le premier étage est vide ou occupé et le dernier étage vide ou occupé. Associé à chaque étage d'information, une bascule d'état indique si l'étage d'information est vide ou occupé. La logique de gestion assure automatiquement le décalage d'information d'un étage occupé vers un étage vide.

Contrairement aux mémoires à structure matricielle, il n'y a plus d'adresse.

7. ACCÈS ALÉATOIRE ET SÉQUENTIEL

Dans une mémoire à structure matricielle, les adresses peuvent se succéder de manière quelconque. On dit que l'accès est *aléatoire.*

Par contre, dans une mémoire à structure linéaire comme une FIFO ou une mémoire à bulle, après avoir lu ou écrit une donnée de rang n, on ne peut accéder qu'à celle de rang suivant. Dans ce cas, l'accès est dit *séquentiel.*

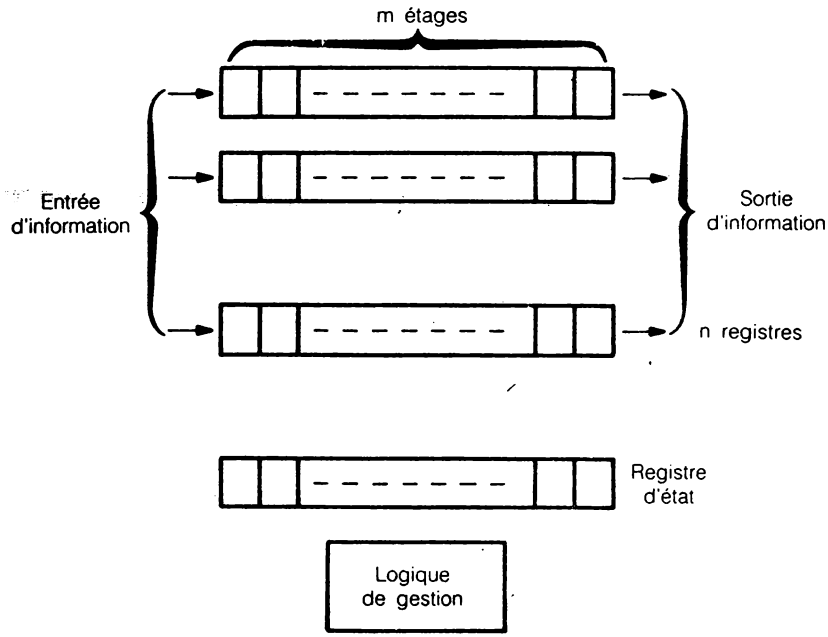


Fig. 15.2. — *Structure d'une mémoire FIFO* $m \times n$.

8. CAPACITÉ ET FORMAT D'UNE MÉMOIRE

La capacité représente le nombre total d'éléments binaires, et le format, l'arrangement de ces eb. Ainsi une mémoire de 4096 eb peut être organisée en 4096 x 1, c'est-à-dire, 4096 groupes ou *mots* de 1 eb ou encore en 1024 x 4 c'est-à-dire 1024 mots de 4 eb. Dans le premier cas, on lit ou écrit un seul eb à la fois tandis que dans le second on traite simultanément 4 eb à la fois. Actuellement il existe beaucoup de mémoires au format de 8 eb car les circuits de calcul associés aux mémoires traitent des quantités de 8 ou 16 eb. On compte la capacité des mémoires en multiple de 1024, représentés par K pour kilo ($1024 \simeq 1000 = 1$ kilo). Ainsi, 4096 mots de 1 eb deviennent $4\,K \times 1$.

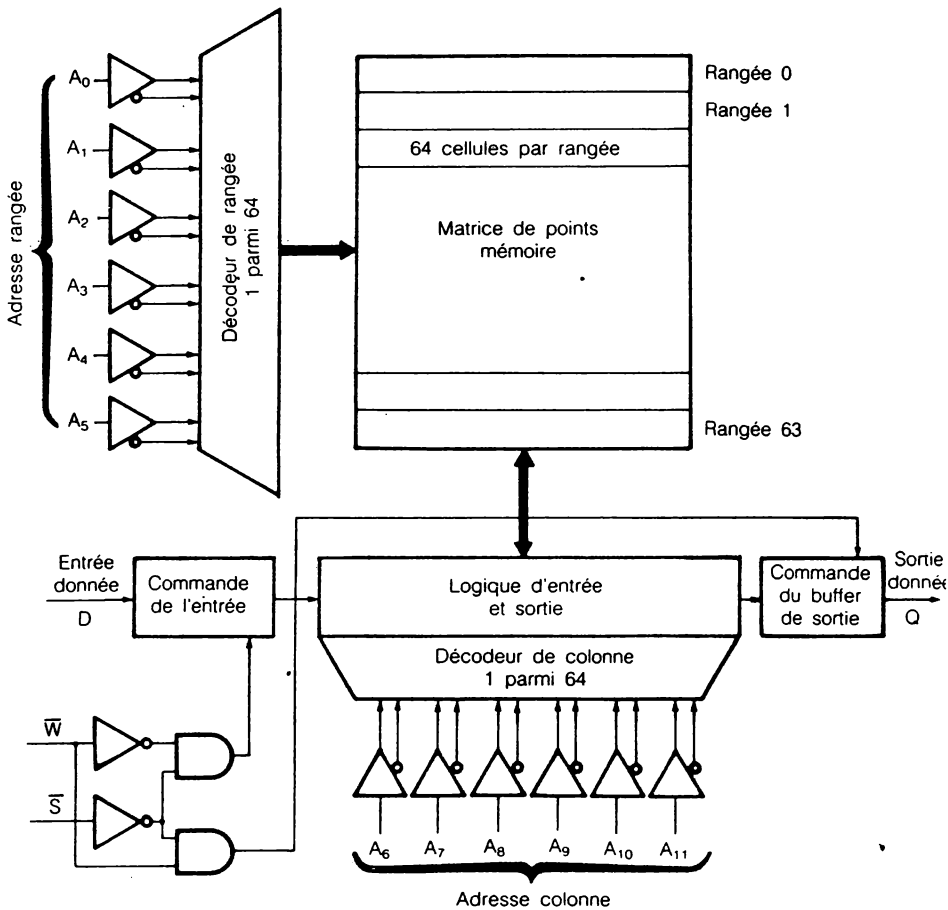


Fig. 15.3. — *Organisation d'une mémoire de 4096 eb.*

La figure 15.3 donne le schéma synoptique d'une mémoire 4096 × 1. Elle comprend :

- Une matrice de 4096 points mémoire organisées en 64 rangées de 64 points chacune.
- Deux décodeurs, l'un pour les rangées, l'autre pour les colonnes. Le premier décode les 6 eb d'adresse A_0^5 (1 parmi 64) et l'autre les 6 eb d'adresse A_6^{11} (1 parmi 64).
- Un signal de commande de sélection $\overline{S}$ (Select). Il valide une opération d'écriture ou de lecture lorsqu'il vaut 0 et bloque la mémoire (interdiction d'écriture et de lecture, sorties en mode haute impédance) lorsqu'il vaut 1.
- Un signal de commande d'écriture $\overline{W}$. Si $\overline{S} = 0$, et $\overline{W} = 0$, la mémoire écrit dans la cellule désignée l'information présente sur l'entrée des données D. Si $\overline{S} = 0$, $W = 1$, la mémoire fournit sur la sortie Q, l'état de la cellule désignée.

9. VITESSE D'UNE MÉMOIRE

Deux paramètres essentiels caractérisent la vitesse d'une mémoire. Le temps d'accès représente le temps qui sépare une demande de lecture de l'obtention de l'information désirée. Le temps de cycle représente l'intervalle qui sépare deux demandes successives de lecture ou d'écriture. Le temps de cycle est égal ou supérieur au temps d'accès.

10. CONSOMMATION D'UNE MÉMOIRE

Les boîtiers à haute intégration ne peuvent dissiper guère plus d'un demi-watt, et les constructeurs ont toujours cherché à tenir dans cette limite quel que soit le nombre d'éléments binaires intégrés dans le boîtier. Chaque augmentation de la densité s'accompagne donc d'une diminution correspondante de la consommation d'une cellule. La consommation du boîtier est donnée soit en milliwatt, soit en milliampères. Certains boîtiers se caractérisent par plusieurs paramètres de consommation détaillés ci-après.

10.1. État actif

La consommation à *l'état actif* (Operating), correspond au cas où la mémoire est en écriture ou en lecture. Le courant moyen consommé par le boîtier dépend de son rythme de travail, c'est-à-dire du temps de cycle. Il existe en effet des courants transitoires dont l'effet se fait d'autant plus sentir sur la consommation que le temps de cycle est court.

10.2. État passif

La consommation à *l'état passif* (standby), correspond au cas où le boîtier n'est pas activé. Ce courant est beaucoup plus bas que le précédent. Cette caractéristique est très utile car dans un ensemble de boîtiers mémoire, seuls quelques uns sont actifs à un moment donné, tous les autres restant inactifs. Plus la capacité mémoire croît, plus le nombre de boîtiers actifs diminue par rapport à l'ensemble et la consommation moyenne se rapproche de la consommation à l'état passif. Les boîtiers qui consomment moins à l'état passif, quand ils ne sont pas sélectionnés sont dits "automatic power down".

10.3. État de veille

La consommation à *l'état de veille* concerne des mémoires qui conservent leur information sous une tension réduite, par exemple 3 *V* au lieu de 5 *V*. Cette possibilité permet, en cas de coupure temporaire de l'alimentation, d'alimenter les mémoires par des batteries et ceci évite de recharger toutes les mémoires après un tel incident. Cependant, pendant que la tension est réduite, la mémoire ne peut fonctionner ni en lecture ni en écriture. Son rôle consiste simplement à sauvegarder l'information. Le courant consommé dans ce cas est encore plus bas que dans le deuxième cas.

Les deux derniers modes de fonctionnement à puissance réduite concernent les mémoires MOS. Cependant, il apparait des mémoires bipolaires à consommation réduite.

11. ALIMENTATION

Aujourd'hui toutes les mémoires fonctionnent avec une seule tension de + 5 *V*, à l'exception des EPROM, EEPROM qui nécessitent une tension élevée de 21 *V* pour la programmation. Cependant certaines EEPROM n'ont besoin que du 5 *V*, générant en interne la haute tension de programmation à partir du 5 *V*. Enfin certaines mémoires vives disposent d'une pile au lithium incorporée dans le boîtier, d'une durée de vie garantie de 5 ou 10 ans.

12. LE BROCHAGE

12.1. Normalisation du brochage

Le brochage des boîtiers concerne le nombre de broches et la répartition des signaux sur celles-ci. Chaque fois qu'un nouveau type de mémoire apparait, le problème de la compatibilité du brochage se pose. Les utilisateurs sont extrêmement intéressés par sa normalisation car cela permet de placer sur les mêmes circuits imprimés des boîtiers provenant de plusieurs constructeurs différents. Certains utilisateurs se refusent à employer un composant n'ayant qu'une seule source d'approvisionnement.

Un comité américain appelé JEDEC, réunissant les constructeurs de composants, établit des projets de normes de brochage pour les boîtiers futurs. C'est pourquoi le brochage de certains boîtiers est qualifié de "JEDEC Standard Pinout" ou de "JEDEC approved". Quand le brochage n'est pas normalisé JEDEC, mais que le constructeur a réussi à prendre une position dominante sur le marché pour un type de produit, le brochage est

décrit comme étant un standard ou encore un brochage reconnu standard par le monde industriel, etc... Il y a aussi la solution de ne rien dire quand le boîtier eet vraiment particulier à un constructeur. La compatibilité du boîtier avec des boîtiers de famille différente est mentionnée: par exemple, compatibilité entre le brochage de PROM et de ROM, ou entre des EPROM et des ROM, ce qui permet d'implanter des boîtiers ROM sur les mêmes cartes lorsque la mise au point avec les PROM et les EPROM est terminée (par quantité, les ROM sont moins chères que les PROM et les EPROM).

La conception du brochage a paru une question suffisamment importante pour que le constructeur MOSTEK introduise des mémoires au format octet. Avec des brochages compatibles entre RAM, EPROM, ROM, on peut réaliser des cartes mémoires très modulaires, avec des proportions variables de ces différents types de boîtiers. La normalisation s'effectue alors sur des boîtiers à grand nombre de broches, 24 ou 28. Les circuits imprimés conçus pour un type de boîtier RAM par exemple, restent valables si les données, étant stabilisées, sont placées en EPROM ou ROM.

12.2. Réduction du nombre de broches

Réduire le nombre de broches d'un boîtier permet d'en augmenter la densité d'implantation sur des circuits imprimés. La meilleure densité s'obtient avec des mémoires au format de 1 eb. Prenons par exemple une mémoire de 2048 mots de 8 eb. Il faut: 8 broches d'entrée d'information, 8 broches de sorties, 11 broches d'adresse ($2^{11} = 2048$), soit 27 broches sans compter les alimentations et les signaux de gestion. Une mémoire de même capacité mais organisée en 16 384 mots de 1 eb ne nécessitera que deux broches pour l'entrée et la sortie et 14 broches d'adresse ($2^{14} = 16\,384$), soit 16 broches en tout. Pour réduire le nombre de broches, les constructeurs ont utilisé trois méthodes:

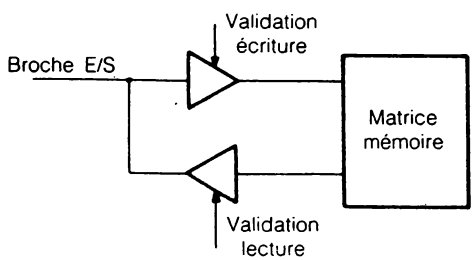


Fig. 15.4. — *Entrées et sorties communes.*

— Réunir sur la même broche d'information d'entrée et l'information de sortie, en tirant parti du fait que l'on ne réalise jamais une écriture et une lecture en même temps. Cela implique un schéma d'entrée-sortie utilisant des portes trois états (fig 15.4). Cette méthode s'emploie systématiquemet pour les mémoires au format de 4 eb ou 8 eb par mot.

— Diviser l'adresse en deux parties et la mémoriser successivement dans le boîtier en utilisant les mêmes broches d'entrée. L'adresse globale rentre sur un multiplexeur extérieur et deux signaux RAS (Row Address Strobe: validation adresse rangée) et CAS (Column Address Strobe: validation adresse colonne) mémorisent ces deux moitiés d'adresse dans deux registres internes au boîtier mémoire. La figure 15.5 montre l'adresse A_0^{15} à 16 eb d'une mémoire à 65 536 eb divisée en A_0^7 et A_8^{15} et rangée dans deux registres. Cette méthode s'emploie systématiquement dans les mémoires dynamiques de grande capacité.

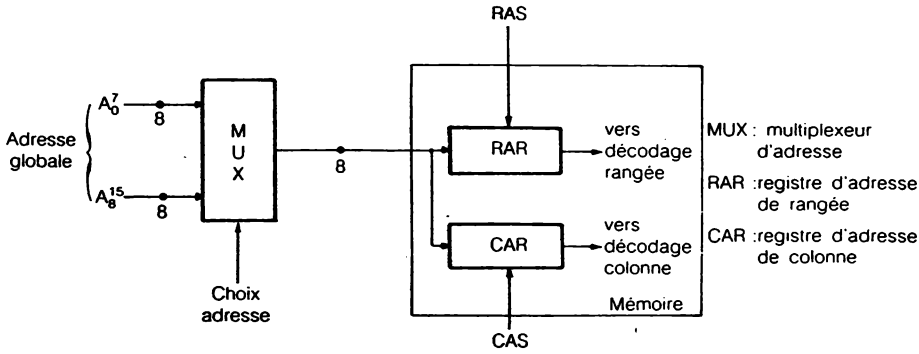


Fig. 15.5. — *Multiplexage de l'adresse*

— Réduire le nombre de tensions nécessaires au boîtier. On est passé de trois tensions à une seule.

13. LES PARAMÈTRES ÉLECTRIQUES ET TEMPORELS

Les mémoires sont réellement des circuits très complexes et leurs caractéristiques électriques et temporelles remplissent de pleines pages de catalogues. On retrouve les paramètres électriques classiques définis au chapitre 5 comme V_{IH}, V_{IL}, V_{OH}, V_{OL}, I_{IH}, I_{IL}, I_{OH}, I_{OL}.

Pour les mémoires bipolaires, la compatibilité avec les circuits TTL ne pose pas de problèmes alors qu'il peut y en avoir avec des mémoires MOS.

Par exemple, un V_{IH} minimum plus élevé que 2,7 V implique une commande par une porte à collecteur ouvert avec résistance de charge pour obtenir un niveau " 1 " élevé. De même, les courants I_{OH}, I_{OL} des mémoires MOS sont faibles et l'on ne peut connecter qu'un nombre restreint de portes TTL (2 portes standard par exemple).

Les paramètres temporels sont nombreux dans les mémoires. On retrouve les paramètres comme la durée d'impulsion, le temps de préaffichage et le temps de maintien. Mais il y a aussi une foule de paramètres indiquant des relations de phases nécessaires entre signaux, dont les noms changent d'un constructeur à l'autre (delay time : temps de retard, overlap time : temps de recouvrement, recovery time : temps de récupération, etc...). Comme les circuits qui dialoguent avec les mémoires possèdent eux aussi des paramètres dynamiques, le respect de tous ces paramètres constitue une entreprise difficile. Des fonctionnements erratiques s'expliquent parfois par la non prise en compte de certains paramètres dans les montages.

14. COMPATIBILITÉ ENTRE MÉMOIRES

Lorsqu'une mémoire connait un certain succès auprès des utilisateurs, les constructeurs autres que le constructeur initial se mettent à la fabriquer. On a alors des produits de seconde source ou des produits compatibles. Le premier niveau de compatibilité, le plus évident, est constitué par le brochage. Mais cela ne suffit pas et le concepteur qui utilise des produits compatibles de constructeurs différents doit s'assurer qu'il y a compatibilité au niveau plus fin des paramètres électriques et temporels.

15. INTÉGRITÉ DE L'INFORMATION

L'augmentation incessante de la capacité des mémoires (un facteur quatre à chaque génération) va de pair avec une réduction de la taille des éléments constituant la cellule. Pour les cellules dynamiques, la capacité

électrique de stockage diminue et par conséquent, la quantité d'électricité stockée. Cette faible quantité rend la mémoire plus sensible aux conditions extérieures. Des particules alpha, émises par le plastique de moulage peuvent détruire des éléments binaires. Les cellules sont également plus sensibles à l'état des cellules voisines (effet de proximité et de configuration), aux parasites d'alimentation. Finalement, la densité accrue se paie par une fragilité plus grande. Pour que l'information reste sûre malgré ces aléas, les utilisateurs ont dû introduire des protections sur l'information, en particulier par code de Hamming (voir chap. 2). L'information ainsi ajoutée n'est pas négligeable (6 eb pour 16, 7 pour 32, 8 pour 64).

16. DESCRIPTION DU FONCTIONNEMENT DES MÉMOIRES

Une mémoire fonctionne selon quelques modes : cycle de lecture, cycle d'écriture, cycle de lecture-modification-écriture, etc... La description s'opère cycle par cycle à l'aide de diagramme des temps indiquant les relations de phase entre les signaux et de tableaux précisant la valeur des paramètres temporels. Le constructeur ne fournit plus, comme pour un MSI, le schéma détaillé, mais un simple schéma bloc très général. L'utilisateur doit

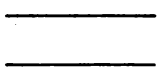
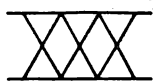
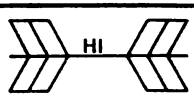
Signal	Entrées	Sorties
	Doit être stable	Sera stable
	Peut changer du niveau haut au niveau bas	Changera du niveau haut au niveau bas
	Peut changer du niveau bas au niveau haut	Changera du niveau bas au niveau haut
	Tout changement autorisé	Changement d'état non connu
	Symbolisme non utilisé	État haute impédance entre deux états basse impédance

Fig. 15.6. — *Symboles utilisés dans les diagrammes de temps*

se pénétrer des contraintes à appliquer aux signaux pour concevoir un système à mémoire. La figure 15.6 donne les symbolismes utilisés couramment dans les diagrammes de temps.

17. QUELQUES QUALIFICATIFS COURANTS CONCERNANT LES MÉMOIRES

Dans les descriptions concernant les mémoires, on trouve des indications précisant leur fonctionnement. Citons les plus courants :

- Fully static memory (mémoire complètement statique) ou fully static operation (exploitation entièrement statique) : indique que les signaux de gestion d'une mémoire statique peuvent être des états. La position des fronts de descente ou de montée des signaux de gestion entre eux peut être quelconque dans le temps.
- Edge activated ou edge triggered (activation par front ou déclenchement sur front) : ceci s'applique aux mémoires statiques pour lesquelles la phase relative des fronts des signaux de commande est primordiale. Ceci s'oppose au mode d'exploitation "fully static" examiné précédemment.
- Registered memory (mémoire avec registre) : il s'agit d'une mémoire avec un registre pour mémoriser l'information lue.
- Unlatched memory (mémoire sans registre) : il s'agit d'une mémoire sans registre de sortie.
- No pull up resistor required : il s'agit d'une mémoire commandable par des circuits TTL classiques, sans qu'il soit besoin d'utiliser un TTL à collecteur ouvert et une résistance de charge.

18. RÉALISATION DE MÉMOIRES DE GRANDE CAPACITÉ

18.1. Assemblages de boîtiers à structure matricielle

Sauf exception, un seul boîtier de mémoire ne suffit pas dans une application, soit parce qu'il manque des eb accessibles en même temps, soit parce qu'il manque des mots ou encore pour ces deux raisons à la fois.

Dans le premier cas, on augmente la *longueur* ou la *taille* du mot en plaçant côte à côte des boîtiers et en les commandant (lecture ou écriture) en même temps. Dans le second cas, on juxtapose des rangées de boîtiers et l'on commande une seule rangée à la fois. Les deux modes d'extension se combinent également. La figure 15.7 donne le principe d'une extension de la taille des mots et du nombre de mots pour obtenir une mémoire de 16 $K \times 8$ à partir de boîtiers 1 $K \times 4$.

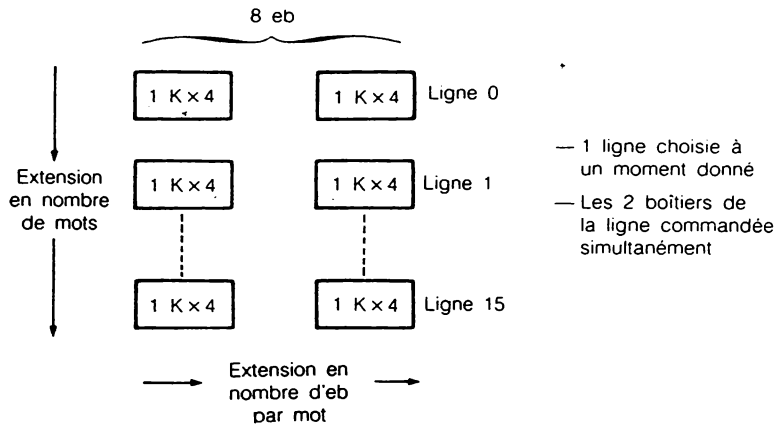


Fig. 15.7. — *Principe de l'extension de la capacité d'une mémoire*

18.2. Mémoire FIFO de grande capacité

Étendre une mémoire FIFO en nombre de mots et en nombre d'eb revient à augmenter la longueur des registres à décalage pour le premier point et à juxtaposer des boîtiers gérés en parallèle pour le second point. La figure 15.8 montre le schéma de principe d'une mémoire FIFO de 192 mots de 8 eb constituée de boîtiers 64 mots de 4 eb.

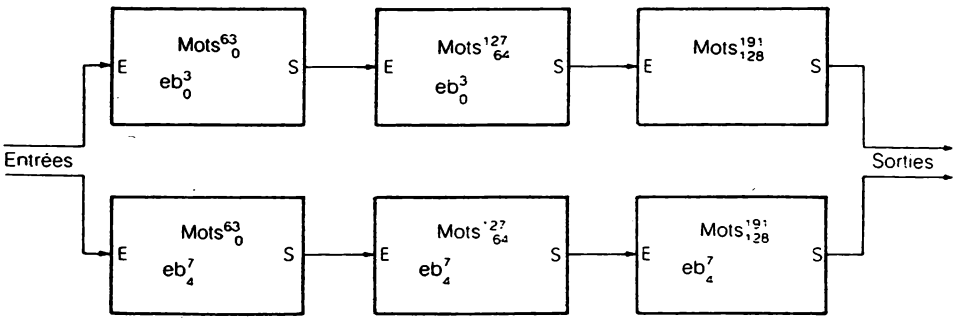


Fig. 15.8. — *Mémoire FIFO de 192 × 8*

18.3. Précaution de réalisation des mémoires de grande capacité

L'assemblage de boîtiers pour constituer une mémoire de grande capacité n'obéit pas seulement à des critères d'organisation logique comme ceux mentionnés au paragraphe précédent. Les constructeurs de boîtiers donnent des conseils pour réaliser ces cartes imprimées, concernant :

- Le dessin lui-même du circuit imprimé de façon à éviter des couplages entre signaux.
- La façon de découpler les alimentations au niveau de la carte, au niveau des boîtiers (calcul des valeurs, emplacements recommandés).
- L'organisation des matrices avec les boîtiers.
- La position des boîtiers d'interface par rapport aux matrices de boîtiers.
- Les précautions à prendre pour éviter des réflexions parasites sur les lignes entre les boîtiers : adaptation par impédance. Ces réflexions, par superposition aux niveaux stables risquent de faire basculer de manière inopportune les entrées des circuits.

Tous ces conseils résultent d'expériences pratiques qu'il ne faut pas négliger si l'on ne veut pas retrouver et résoudre des problèmes déjà connus par d'autres.

19. GÉNÉRALITÉS SUR LES CYCLES DE FONCTIONNEMENT DES MÉMOIRES

Les mémoires fonctionnent selon des cycles : cycle de lecture, cycle d'écriture, etc... Nous donnons dans ce paragraphe les propriétés générales des cycles existants.

19.1. État inactif, état de veille

Dans l'état inactif, la mémoire alimentée normalement n'est pas sélectionnée. On ne peut ni lire ni écrire. Les commandes de lecture et écriture envoyées sur un boîtier non sélectionné n'agissent pas. La sortie, en mode haute impédance dans ce cas permet la connexion en ET câblé (logique positive) avec les autres boîtiers. C'est l'état le plus courant d'un boîtier.

Ainsi, dans la figure 15.7, seuls deux boîtiers sur les trente-deux sont sélectionnés à un instant donné, tous les autres restant dans l'état inactif.

Dans l'état de veille, la mémoire alimentée sous tension réduite avec des batteries ne peut être ni lue, ni écrite. Elle se contente de conserver son information.

19.2. Cycle de lecture

C'est le plus simple des cycles (fig 15.9). La procédure consiste à :

- Établir l'adresse de la cellule ou du mot choisi.
- Sélectionner le ou les boîtiers correspondants.
- Afficher la fonction lecture (c'est l'inverse de la fonction écriture, une seule commande servant aux deux fonctions).
- La sortie quitte le mode haute impédance et l'information apparait au bout du temps d'accès t_a. La sortie reprend le mode haute impédance dès que le boîtier n'est plus sélectionné.

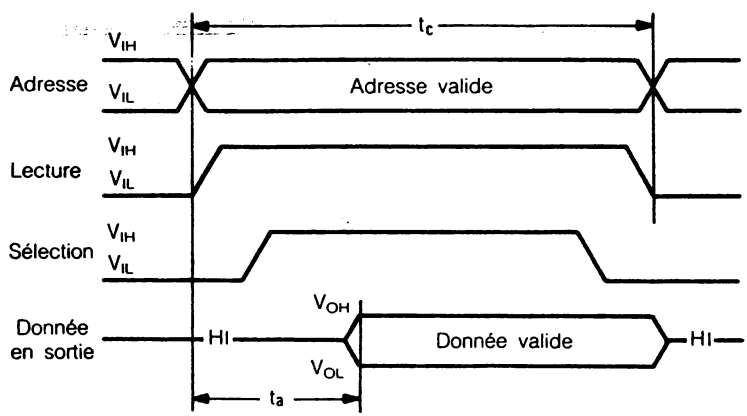


Fig. 15.9. — *Cycle de lecture*

- Une nouvelle interrogation peut reprendre, un temps t_c (temps de cycle après la première).

Le diagramme de la figure 15.9 indique au niveau des lignes d'adresses qu'un eb A_i de l'adresse peut passer du niveau V_{IH} à V_{IL} ou de V_{IL} à V_{IH} et au niveau de la sortie, qu'un eb passe de l'état V_{HI} (haute impédance) soit à l'état V_{OH} (basse impédance), soit à l'état V_{OL} (basse impédance).

Pour une mémoire dynamique, on présente l'adresse en deux parties : d'abord l'adresse ligne, puis l'adresse colonne. Deux signaux RAS (Row Address Strobe : échantillonnage de rangée) et CAS (Column Adress Strobe : échantillonnage de colonne) mémorisent ces deux adresses respectivement. L'information apparaît en sortie pendant CAS. Les temps d'accès se comptent par rapport aux fronts montants RAS et CAS (figure 15.10). Les catalogues de constructeurs donnent les conditions de phase à respecter entre les différents signaux présentés sur les entrées, par exemple RAS et CAS par rapport aux adresses ainsi que les durées des différents signaux.

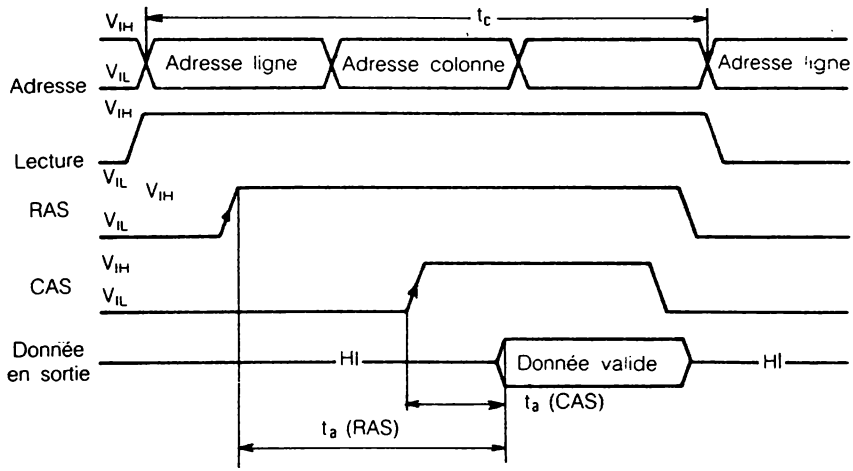


Fig. 15.10. — *Cycle de lecture d'une mémoire dynamique*

19.3. Cycle d'écriture

Ce cycle ne concerne que les mémoires vives (fig 15.11). La procédure consiste à :

- Établir l'adresse (en une fois pour une mémoire statique, en deux fois pour une mémoire dynamique).
- Sélectionner le boîtier.
- Établir la donnée sur l'entrée.
- Établir une impulsion d'écriture (inverse de la lecture).

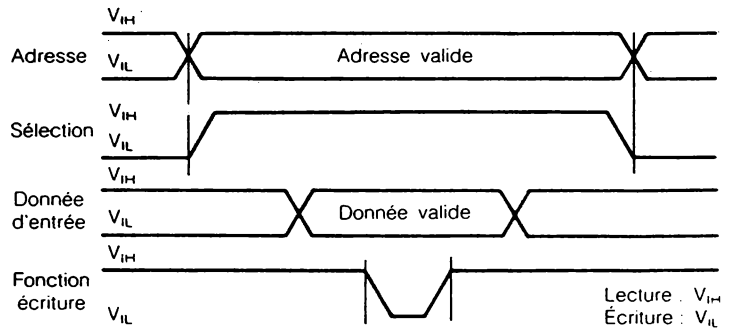


Fig. 15.11. — *Cycle d'écriture.*

19.4. Cycle de lecture – Modification – Écriture

Ce cycle ne concerne que les mémoires vives. Il s'emploie lorsque l'on veut dans le même cycle lire une donnée à une adresse déterminée, modifier la donnée et l'inscrire à la même adresse. Le même résultat s'obtient également avec deux cycles successifs de lecture et d'écriture, mais plus lentement que dans un cycle global de lecture-modification-écriture, car il faut alors présenter l'adresse deux fois. La figure 15.12 montre la procédure :

- Établir l'adresse.
- Sélectionner le boîtier.
- Afficher la fonction lecture.

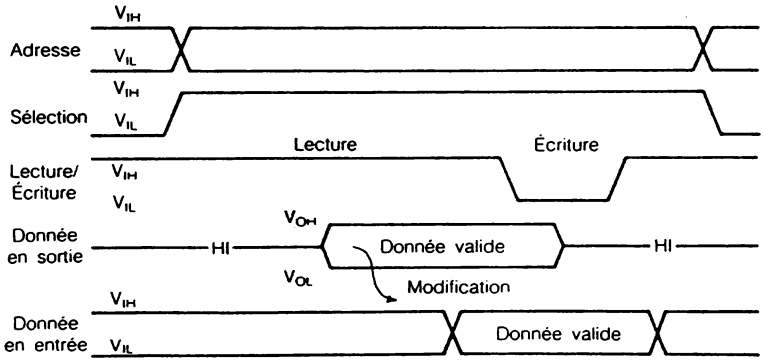


Fig. 15.12. — *Cycle de lecture – Modification – Écriture*

- La donnée apparait en sortie au bout du temps d'accès.
- Le système l'utilise et présente une nouvelle valeur qu'il applique en entrée.
- Appliquer un signal d'écriture.

Dans une mémoire dynamique, ce qui précède reste vrai, l'adresse étant présentée en deux fois.

19.5. Cycles de lecture, écriture en mode page (fig. 15.13)

Ces cycles n'existent que pour les mémoires vives dynamiques. Si une suite d'opérations d'écriture ou de lecture ou de lecture-modification-écriture ont lieu avec la même adresse ligne, il n'est pas nécessaire de fournir à chaque fois l'adresse ligne. On donne donc une seule fois l'adresse ligne, puis les adresses colonne successives. Ceci permet de gagner du temps dans les échanges. Ce mode est appelé page parce que l'on conserve fixe une partie de l'adresse qui permet de s'adresser à une même portion de mémoire. Ainsi, avec une mémoire de 64 K mots de 8 eb, réalisée avec des boîtiers 64 $K \times 1$, l'adresse A_0^{15} se subdivise en A_0^7 pour les colonnes et A_8^{15} pour les lignes. En conservant fixe une adresse ligne ($A_8^{15} = i$), on peut accéder à une zone comprise entre :

($A_8^{15} = i$), ($A_0^8 = 00$) et ($A_8^{15} = i$), ($A_0^8 = FF$) en héxadécimal

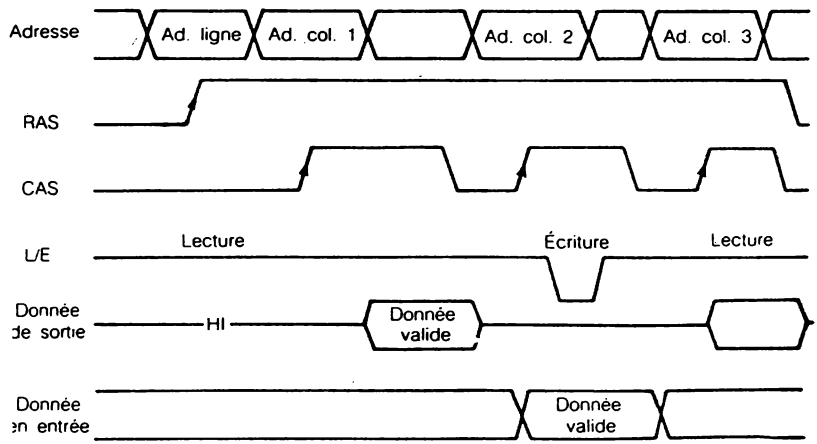


Fig. 15.13. – *Cycles en mode page*

La zone mémoire comprise entre $A_0^8 = 00$ et $A_0^8 = FF$ constitue une page de 256 mots (en décimal).

19.6. Cycle de rafraîchissement

Une mémoire dynamique fonctionne en deux temps. D'abord lecture de tous les eb d'une rangée (ou ligne) et réinscription de tous ces eb. Puis lecture ou écriture d'un seul eb de la rangée sélectionné grâce à l'adresse colonne. On voit donc que la lecture ou l'écriture d'un eb permet de rafraîchir tous les eb de la rangée auquel il appartient, c'est-à-dire de restaurer le niveau de la charge stockée dans la capacité de structure. La période de rafraîchissement est habituellement de 2 *ms* ou 4 *ms*. Mais dans un système, toutes les cases mémoires ne sont pas lues ou écrites de manière régulière, de sorte que l'on ne peut pas garantir qu'une case soit visitée régulièrement avec une période inférieure à 2 *ms* ou 4 *ms*. Pour être sûr que cette mémoire soit bien restaurée, on doit prévoir des cycles de rafraîchissement à intervalles réguliers indépendants des cycles de lecture et écriture.

Compte tenu du fonctionnement décrit plus haut, le rafraîchissement intervient pendant la première partie du cycle d'opération. Pour cette raison, le cycle de rafraîchissement utilise seulement l'adresse de rangée et le signal RAS (fig 15.14).

Pendant que la mémoire effectue des cycles de rafraîchissement, elle n'est pas disponible pour des cycles de lecture ou d'écriture. Sur un cycle de 2 *ms*, le temps ainsi consommé pour cette fonction est de l'ordre de 2 %.

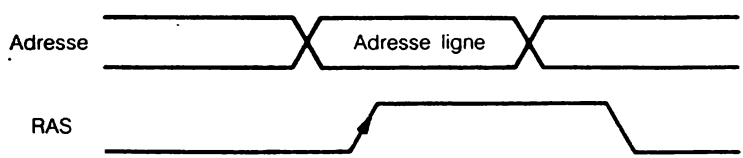


Fig. 15.14. — *Cycle de rafraîchissement*

La priorité doit être donnée, dans le système de gestion de la mémoire, aux cycles de rafraîchissement sur les cycles de lecture et d'écriture, sinon l'information risque de se détruire. Il en résulte que l'organe qui interroge la mémoire doit être capable d'attendre une réponse dans un délai variable, ce qui complique également l'interface, une simple solution synchrone avec temps fixe ne convenant pas.

19.7. Programmation

Ceci s'applique uniquement aux ROM. L'utilisateur définit sur des cartes perforées ou autre support informatique, le contenu de la mémoire. A partir de ces données, le constructeur réalise les métallisations correspondant aux 1 ou 0.

19.8. Cycle de programmation

Ce cycle concerne les mémoires PROM et EPROM. Dans une mémoire PROM, ce cycle détermine les signaux et les tensions à appliquer pour faire fondre sélectivement les fusibles et dans une mémoire EPROM, les signaux pour stocker une charge. Ces cycles mettent en œuvre des signaux à tension élevés (environ 20 *V*) et des mesures analogiques des tensions de sortie pour s'assurer de la bonne programmation.

19.9. Effacement

Ceci ne concerne que les mémoires EPROM qui sont soumises à un flux de rayons ultra-violets pour effacer les charges stockées. Une exposition d'un quart d'heure au moins est nécessaire.

19.10. Récapitulation des cycles

La figure 15.15 récapitule tous les cycles existants pour toutes les mémoires.

	Mémoires					
Cycles	RAM statique	RAM dynamique	ROM	PROM	EPROM	FIFO
État inactif	X	X	X	X	X	
État de veille	X	X	Non volatile	Non volatile	Non volatile	
Lecture	X	X	X	X	X	X
Écriture	X	X				X
Lecture - modification écriture	X	X				
Mode page		X				
Rafraichissement		X				
Programmation par constructeur			X			
Programmation électrique (client)				X	X	
Effacement (U.V.)					X	

Fig. 15.15. — *Récapitulatif des cycles mémoires*

20. LES RÉSEAUX LOGIQUES PROGRAMMABLES

20.1. Réseaux combinatoires

La figure 15.16 montre la structure d'un FPLA (Field Programmable Logic Array : réseau logique programmable par l'utilisateur). Il comprend :

— Des amplificateurs qui délivrent les variables d'entrée sous forme directe et inverse.

— Des portes ET qui calculent le produit de toutes les variables d'entrées, directes et inverses.

— Des portes OU, qui reçoivent toutes les sorties des portes ET.

Ce réseau permet par sa constitution d'obtenir des fonctions booléennes exprimées en forme de sommes de produits. Il est program-

mable, car les liaisons entre les variables (directes, inverses) et les portes ET, et entre les portes ET et OU dépendent de microfusibles que l'on peut laisser intacts ou détruire (comme dans une mémoire PROM).

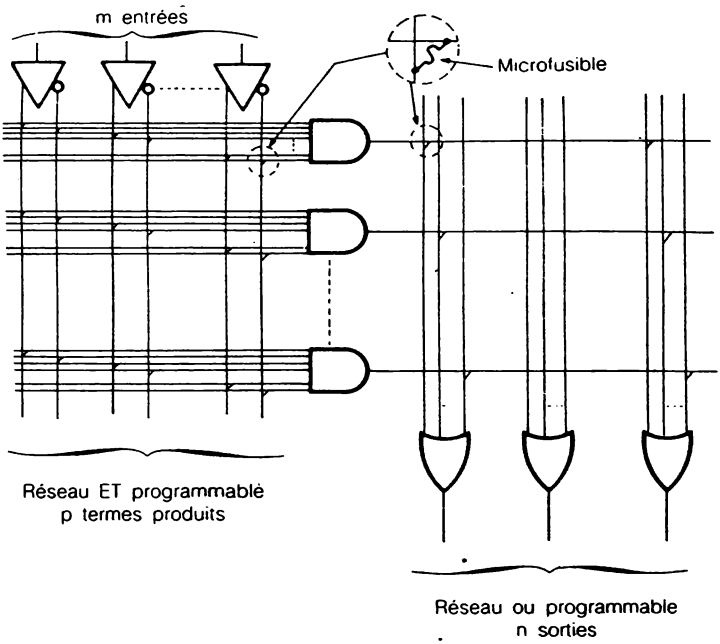


Fig. 15.16. — *Constitution d'un FPLA.*

La figure 15.17 donne une représentation condensée de la matrice ET, OU et la programmation des fusibles pour élaborer quatre fonctions booléennes :

- $f_1 = x\bar{y}t + \bar{x}zt + xyz\bar{t}$
- $f_2 = \bar{x}zt + xyz\bar{t}$
- $f_3 = x\bar{y}t + xyz\bar{t}$
- $f_4 = \bar{x}zt$

On remarque que les portes ET 4 et 5 de la figure 15.17 ont leurs fusibles intacts en entrée et en sortie. Cela n'est pas gênant car elles fournissent les fonctions $x\bar{x}y\bar{y}z\bar{z}t\bar{t}$, toujours nulles. Cette propriété permet de garder en réserve les réseaux correspondants pour éventuellement calculer d'autres fonctions, ou modifier et compléter les fonctions précédemment calculées.

Les FPLA ont été introduits initialement par Signetics en 1975.

En 1978, Monolithic Memories Incorporation proposa des circuits appelés PAL (Programmable Array Logic) programmables également par l'utilisateur, de structure voisine de celle des FPLA, mais avec une connexion fixe entre ET et OU. La figure 15.18 montre la structure d'un PAL combinatoire de type 12 H6, comprenant douze entrées, six sorties directes regroupant deux ou quatre monômes.

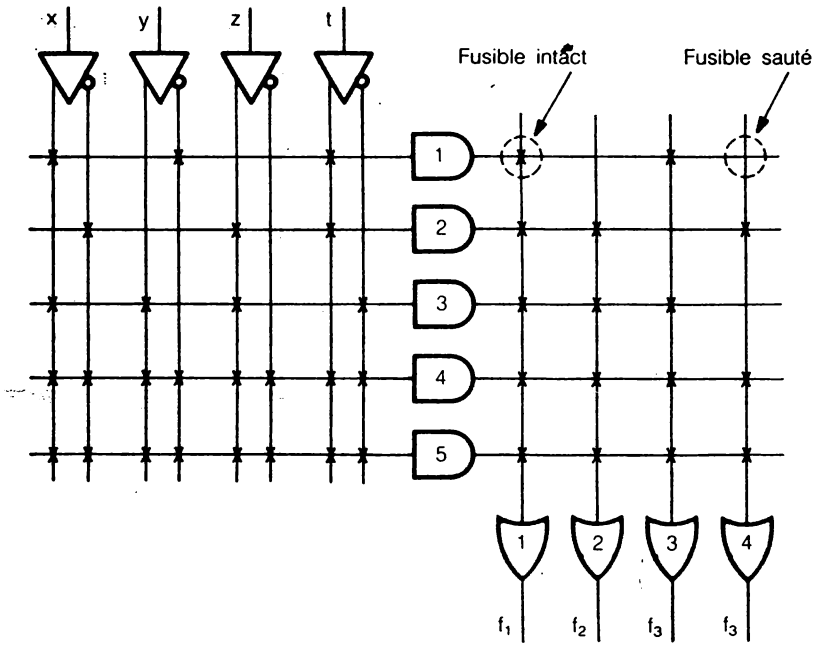
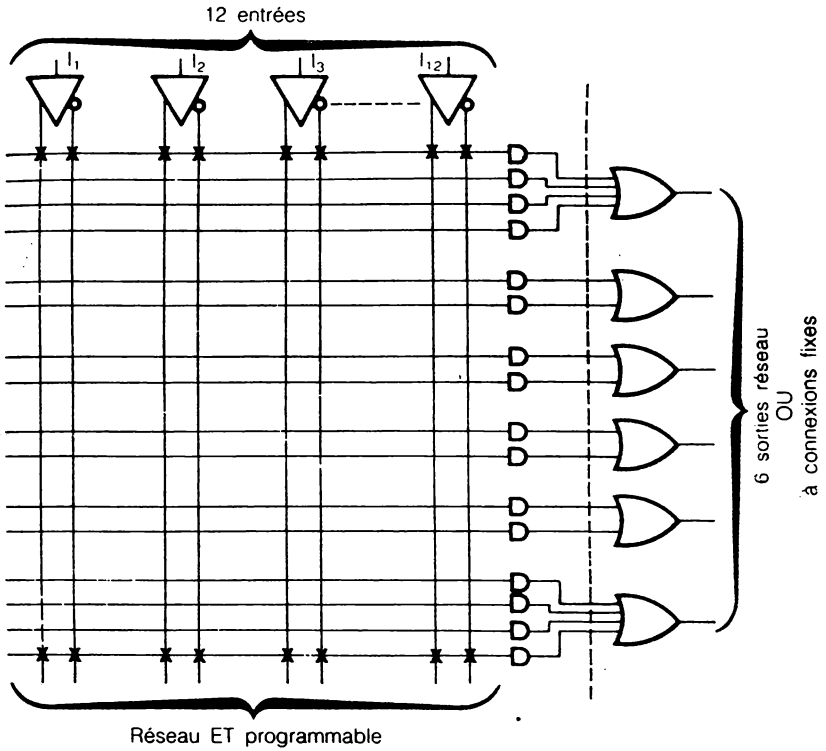


Fig. 15.17. — *Réalisation de fonctions combinatoires avec un FPLA.*



Tous les fusibles n'ont pas été représentés.

Fig. 15.18. — *Structure d'un PAL combinatoire 12 H6.*

La figure 15.19 montre la réalisation des fonctions f_1, f_2, f_3, f_4 décrites plus haut avec un PAL de type 12 H6. Il existe plusieurs types de PAL qui se différencient par les nombres d'entrée, de sorties, de portes ET regroupées sur les portes OU, les sorties trois états rebouclées ou non sur les entrées, la polarité des sorties (directes ou inverses) etc... .

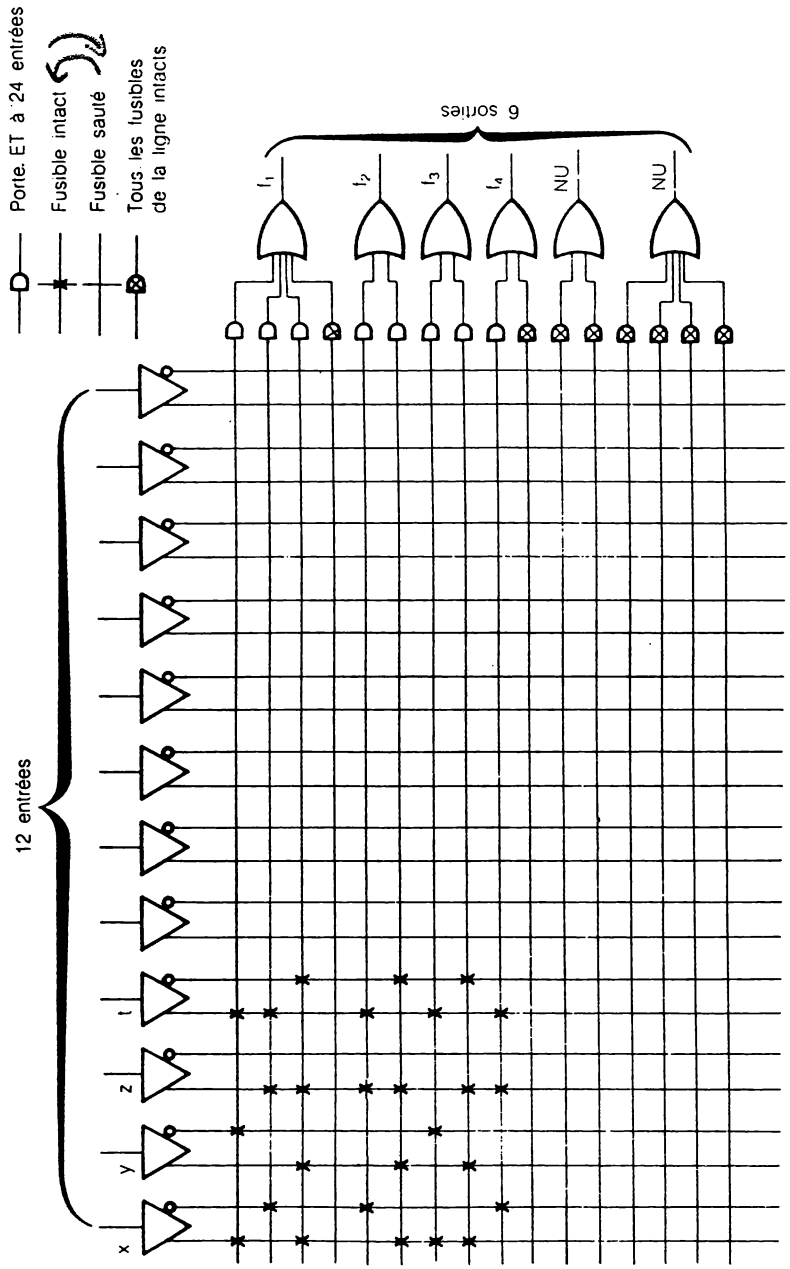


Fig. 15.19. — *Obtention de fonction avec un PAL combinatoire 12 H6.*

20.2. Réseaux séquentiels programmables

La figure 15.20 donne le schéma de principe d'un PAL séquentiel de type 16 R4. Pour ne pas surcharger la figure, les fusibles ne sont pas représentés. Ce PAL comprend :

- Un réseau combinatoire indépendant constitué de portes ET, OU, Inverseur trois états. La sortie de chacun des inverseurs est réinjectée dans le réseau de portes ET pour constituer une nouvelle variable d'entrée. Dans chacun des quatre sous-ensembles de ce type, sept portes ET ayant chacune 32 entrées alimentent une porte OU pour élaborer une fonction de sept monômes maximum. La dernière porte ET commande l'inverseur soit en mode basse impédance si l'on veut exploiter la fonction de sortie dépendant des sept monômes, soit en mode haute impédance, ce qui permet alors de se servir de la sortie comme d'une entrée. Comme les conditions de commande de l'état haute ou basse impédance dépendent de variables, on peut effectuer des échanges bidirectionnels sur les sorties des inverseurs.
- Un réseau combinatoire constitué de portes ET-OU connecté aux entrées *D* des bascules.
- Un réseau séquentiel comprenant des bascules *D*, avec une horloge commune. Les sorties des bascules sont réinjectées dans le réseau de portes ET, et constituent aussi des entrées. On peut donc réaliser des fonctions d'automodification ou de modifications croisées des bascules.
- Les sorties des bascules sont présentées en sortie du boîtier à travers un inverseur trois états, ce qui permet la réunion de plusieurs boîtiers en ET câblé.

Comme indiqué plus haut, on peut élaborer une fonction séquentielle quelconque définie par un tableau de vérité, ou une fonction séquentielle classique, comme un compteur, un registre à décalage ou autre.

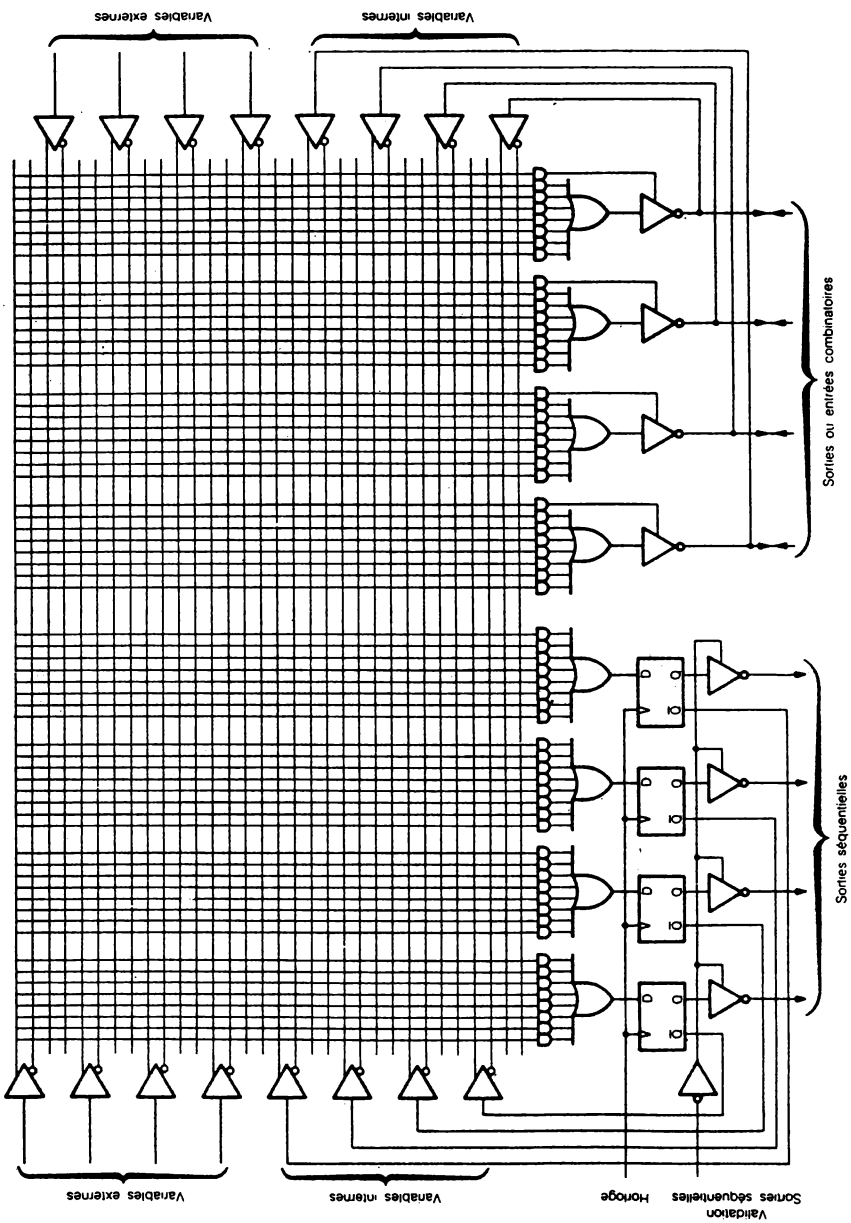


FIGURE 20 : PAL SEQUENTIEL DE TYPE 16 R4

Fig. 15.20. — *PAL séquentiel de type 16 RA.*

ANNEXE DU CHAPITRE 15

1. UNE TRÈS GRANDE DIVERSITÉ DANS LES MÉMOIRES

Les mémoires sont des circuits qui évoluent et se démodent très rapidement. A un instant donné on constate un foisonnement dans les types de mémoire et quelques circuits sur lesquels un consensus s'est établi entre constructeurs pour les produire. Ce peut être alors ce que l'on appelle de "vraies secondes sources" résultant d'un échange de technologie entre deux constructeurs ou des "fausses secondes sources", deux constructeurs réalisant avec leur technologie propre un produit semblable.

Les constructeurs publient des tableaux d'équivalence entre leurs mémoires et les autres (fig. 15.21).

		Références équivalentes des autres constructeurs						
Boîtier	Harris	Fujitsu	Hitachi	Intersil	Intel	Motorola	National	RCA
RAM CMOS 256 × 4	6561	—	—	6561	—	—	74 C 921	—
RAM CMOS 1024 × 4	6514	8414	6148	6514	21 C 14	6514	6514	5114

Fig. 15.21. — *Équivalences de deux boîtiers Harris.*

Il faut cependant être très prudent pour les équivalences. Bien entendu les caractéristiques globales : tension d'alimentation, capacité, organisation, temps d'accès et de cycle sont les mêmes. Mais cela ne suffit pas à rendre des boîtiers interchangeables sur une carte. Il faut examiner avec soin si les

caractéristiques temporelles sont bien respectées, si les consommations sont compatibles ainsi que les entrances ou sortances.

Parmi les principaux constructeurs citons les américains Advanced Micro devices (AMD), General Instrument, Fairchild, Harris, Intel, Intersil, Mostek, Motorola, Monolithic Memories Incorporation (MMI), National Semi conductor (NS), RCA, Synertec, Signetics, Texas Instruments (TI), les japonais Fujitsu, Hitachi, NEC, Toshiba, ainsi que les groupes français associés à des firmes américaines dans le cadre du plan composants Matra/-Harris, Eurotechnique Thomson, Thomson EFCIS/Motorola.

2. UN ÉCHANTILLON DE MÉMOIRES

Les mémoires étant très diverses nous n'avons donné qu'un échantillon de ce qui existe sur le marché. La figure 15.22 montre quelques types de RAM statique MOS, TTL et CMOS, RAM dynamique MOS et mémoire FIFO. On note qu'il y a très peu de mémoires TTL. Cela tient à ce que l'on sait maintenant obtenir avec la technologie MOS des vitesses comparables à celles de la TTL pour une consommation comparable ou moindre. Aussi les mémoires TTL tendent à disparaître au profit des mémoires MOS. On réalise cependant des mémoires bipolaires, très rapides de faible capacité pour des besoins particuliers, comme la mémoire Am 29705 de AMD. Elle sert à constituer des registres pour les microprocesseurs en tranches Am 2900. Sa structure est assez particulière puisqu'elle possède deux accès d'adresse et deux sorties d'information.

Pour les mémoires dynamiques on trouve deux temps d'accès, le premier par rapport au signal de validation de rangée (RAS), et le second, plus court par rapport au signal de validation de colonne (CAS). On note que les mémoires dynamiques les plus anciennes (8192 × 1 et 16384 × 1) sont alimentés par trois tensions au lieu d'une pour la mémoire dynamique la plus récente.

Sauf pour les mémoires TTL, il existe une consommation au repos et en fonctionnement. Pour les mémoires CMOS, la consommation au repos est très basse (1 *m W* ou moins) et la consommation en fonctionnement dépend de la vitesse de travail.

Pour la mémoire FIFO on ne définit pas de temps d'accès mais une vitesse de fonctionnement qui est la fréquence d'entrée ou de sortie de donnée. Le temps d'accès varie selon le remplissage de la mémoire.

Technologie	Constructeur	Référence	Capacité en eb	Organisation mot × eb	Brochage	Temps d'accès max (ns)	Entrees. sorties communes	Consommation maximum du boîtier en mW	Observations
RAM statique MOS	Texas Inst	TMS 4044 – 45	4096	4096 × 1	18	450	non	156/440	1 seule tension +5 V
	Texas Inst.	TMS 4044 – 15	4096	4096 × 1	18	150	non	156/440	
	Texas Inst.	TMS 2147 – 5	4096	4096 × 1	18	55	non	150/900	
	Texas Inst.	TMS 2114 – 15	4096	1024 × 4	18	150	oui	350/500	
	Texas Inst.	TMS 4016	16384	2048 × 8	24	450	oui	450 mW	
	Mostek	MK 2148 – 70	4096	1024 × 4	18	70 ns	oui	150/625	
RAM statique TTL	Texas Inst.	74 S 189 A	64	16 × 4	16	50	non	525	2 accès d'adresse
	Texas Inst.	74 S 201	256	256 × 1	16	65	non	700	
	AMD	Am 29705	64	16 × 4	28	53	non	1050	
RAM statique CMOS	Harris	HM 6561 – 9	1024	256 × 4	18	300	oui	55 μW/22 mW par MHz	registre d'adresse et de sortie
	Harris	HM 6518 – 9	1024	1024 × 1	18	310	non	idem 6561 – 9	idem 6561 – 9
	Harris	HM 6533 – 9	4096	1024 × 4	22	400	oui	1 μW/35 mW par MHz	idem 6561 – 9
	Harris	HM 6543 – 9	4096	4096 × 1	22	400	non	idem 6533 – 9	idem 6561 – 9
RAM dynamique MOS	Texas Inst.	TMS 4108 – 15	8192	8192 × 1	16	150/100	non	20/462	+5 V +12 V −5 V
	Texas Inst.	TMS 4116 – 15	16384	16384 × 1	16	150/100	non	20/462	
	Texas Inst.	TMS 4164 – 15	65536	65536 × 1	16	150/100	non	25/185	+5 V
FIFO TTL	MMI	C 67401 A	256	64 × 4	16	15 MHz	non	800	5 V

Fig. 15.22. — *Mémoires RAM et FIFO.*

Technologie	Constructeur	Référence	Capacité en eb	Organisation mot × eb	Brochage	Temps d'accès max (ns)	Consommation maximum du boîtier en mW	Tensions	Observations
PROM TTL	MMI	6331 – 1	256	32 × 8	16	50	625	5/30/24	Formats existants
	MMI	6389 – 1	8192	2048 × 4	18	70	850	5/30/24	256 × 4, 512 × 4,
	MMI	6381 – 1	8192	1024 × 8	24	90	875	5/30/24	1024 × 4, 2048 × 4.
	MMI	63 LS 241	2048	512 × 4	16	75	350	5/11	32 × 8, 256 × 8, 512 × 8,
	MMI	63 RA 441	4096	1024 × 4	18	80	825	5/11	1024 × 8, 2048 × 8,
	Matra Harris	HM 76321	32768	4096 × 8	24	65	950	5/11	4096 × 8, 8192 × 8.
	Matra Harris	HM 76641	65536	8192 × 8	24	85	950	5/11	
EPROM MOS	Texas Inst.	TMS 2508 – 25	8192	1024 × 8	24	250	125/425	5 V/25 V	
	Texas Inst.	TMS 2516 – 35	16384	2048 × 8	24	350	125/500	5 V/25 V	
	Texas Inst.	TMS 2532	32768	4096 × 8	24	450	125/475	5 V/25 V	
	Texas Inst.	TMS 2564	65536	8192 × 8	28	450	125/800	5 V/25 V	
EEPROM MOS	Général Inst.	ER 5716	16384	2048 × 8	24	300	300	5 V/25 V	
	Général Inst.	ER 5916	16384	2048 × 8	24	300	–	5 V	

Fig. 15.23. — *Mémoires non volatiles: PROM, EPROM, EEPROM.*

Technologie	Constructeur	Référence	Capacité en eb	Organisation mot × eb	Brochage	Temps d'accès max (ns)	Consommation maximum du boîtier en mW	Tensions	Observations
MOS	Texas Inst.	TMS 4732	32768	4096 × 8	24	450	–	5 V	
		TMS 4764	65536	8192 × 8	24	450	–	5 V	
TTL	MMI	6281 – 1	8192	1024 × 8	24	80	775	5 V	
	MMI	6276 – 1	16384	2048 × 8	24	110	950	5 V	
	MMI	6155	2560	512 × 5	18	100	900	5 V	Générateur de 64 caractères ASCII en 7 × 5
	MMI	6172	8064	1152 × 7	24	125	900	5 V	Générateur de 128 caractères ASCII en 7 × 9
	MMI	6290	8064	1152 × 7	24	125	900	5 V	Générateur de 128 caractères à la demande
CMOS	Harris	HM 6312 A 9	12288	1024 × 12	18	200	8 /100	5 V	Adresse et sorties communes

Fig. 15.24. — *Mémoires non volatiles: ROM.*

Les figures 15.23 et 15.24 donnent des tableaux de mémoires non volatiles : PROM, EPROM, EEPROM et ROM. Il y a une très grande diversité de mémoires PROM. PROM et EPROM nécessitent des tensions élevées pour leur programmation 11, 24, 25, 30 *V* mais elles fonctionnent toutes avec une seule tension 5 *V* en lecture. Les EPROM, en technologie MOS ont deux consommation l'une au repos, l'autre au travail. Les EEPROM comportent deux tensions 5 *V*, 25 *V* pour les plus anciennes mais la tendance est de n'avoir qu'une seule tension 5 *V* aussi bien pour la lecture que pour la programmation.

Les ROM existent en technologie TTL et CMOS. Le tableau de la figure 15.24 montre aussi des ROM particulières : les générateurs de caractères pour l'affichage sur des matrices de points lumineux.

3. LA COMPATIBILITÉ "BYTE WIDE" OU COMPATIBILITÉ OCTET

Dans une application on peut passer par le stade d'une maquette en RAM pour la mise au point, puis à une série soit en EPROM ou soit en ROM pour de grandes quantités, si les données sont stables. On peut aussi vouloir réaliser avec le même circuit imprimé, et moyennant des modifications simples par strap ou par interrupteur des cartes supportant aussi bien des mémoires RAM que EPROM ou ROM.

Les constructeurs proposent donc des mémoires au format octet ayant des brochages compatibles. La figure 15.25 montre le cas de boîtiers Mostek. Dans la partie centrale de la figure se trouve un boîtier 28 broches (numérotation de 1 à 28) et un boîtier 24 broches (numérotation de 1 à 24 entre parenthèses). La répartition des fonctions de commandes, adresses et données sur les broches des différents boîtiers des mémoires 4802 (RAM), 37000 (ROM), 2764 (EPROM) est telle qu'elle permet l'interchangeabilité de ces trois types de mémoire sans que le cablage externe en soit affecté.

La masse VSS arrive au même endroit physique : broches 14 (boîtier 28 broches) ou 12 (boîtier 24 broches). L'alimentation VCC est distribuée systématiquement sur deux broches 26 (24) et 28. La connexion VCC ne gêne pas les boîtiers 28 broches car la broche 26 est non câblée pour les mémoires ROM et EPROM.

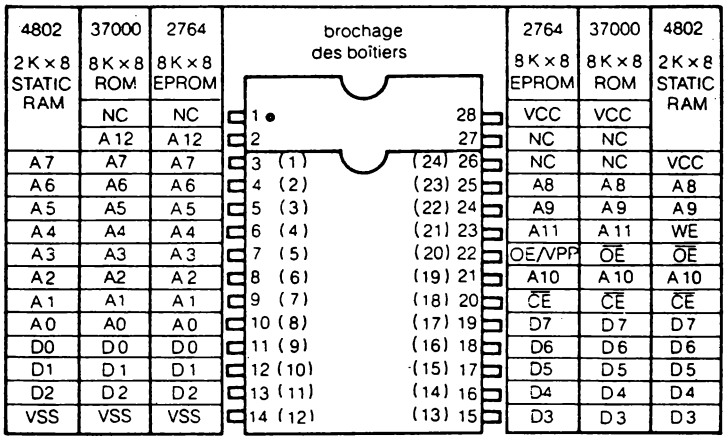


Fig. 15.25. — *Boîtiers à brochages compatibles.*

Les connexions d'information $D0$ à $D7$ se font sur les mêmes emplacements. Il en est de même pour les adresses $A0$ à $A10$. L'adresse $A11$ et l'impulsion d'écriture partagent le même emplacement. C'est la seule broche qui nécessite un câblage différent selon que l'on est équipé de RAM ou de ROM et EPROM.

La sélection du boîtier CE et la validation du boîtier OE se situent au même endroit.

Chapitre 16

APPLICATION DES MÉMOIRES

1. STOCKAGE DES PROGRAMMES ET DES DONNÉES ASSOCIÉES

C'est de loin l'application la plus courante en raison du développement de la logique programmée. C'est aussi celle qui mobilise le plus de boîtiers. Dans les applications utilisant des microprocesseurs, des volumes de 64 *K* mot de mémoire, répartis entre vive, morte ou programmable sont monnaie courante. Les besoins en taille mémoire croissent sans cesse et cela d'autant plus que les programmes sont écrits dans des langages de haut niveau.

2. TABLES DE DONNÉES FIXES

Les mémoires ROM, PROM, EPROM sont idéales pour stocker des tables de données fixes. Ceci augmente la rapidité d'une machine : la consultation d'une table de sinus par exemple, consomme moins de temps que le calcul de la fonction par un développement en série.

3. REMPLACEMENT DE REGISTRES DANS UNE MACHINE

On peut remplacer les registres d'accès à un bloc de calcul par une mémoire ayant deux jeux d'adresse *A*, *B*, servant à choisir le "registre" interne à la mémoire et fournissant les deux opérandes correspondants au circuit arithmétique et logique. Le résultat est inscrit dans la mémoire à l'adresse *B*. Ceci diminue beaucoup le volume du matériel (fig. 16.1), mais augmente le temps de calcul, les mémoires ayant des temps d'accès supérieurs à celui des registres. Cette technique est utilisée dans les processeurs en tranches.

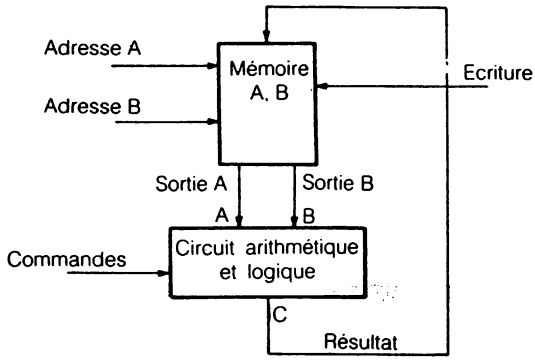


Fig. 16.1. — *Bloc de calcul avec mémoire.*

4. MÉMOIRE D'INTERFACE ENTRE SOUS-ENSEMBLES

Une mémoire peut servir d'interface entre deux sous-ensembles (fig. 16.2). Elle porte le nom de mémoire tampon (ou buffer). Comme dans

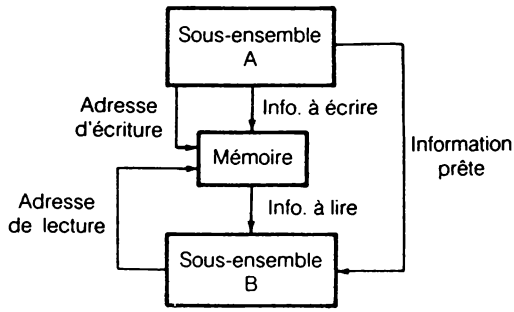


Fig. 16.2. — *Interface réalisé par une mémoire tampon.*

le cas du registre tampon, il doit y avoir coordination entre les deux sous-ensembles. Citons les techniques les plus couramment employées:

- Le système *A* remplit la mémoire et prévient le système *B* quand elle est pleine. *A* n'agit plus sur la mémoire et *B* exploite son contenu. Quand *B* a fini, il prévient *A* que la mémoire est redevenue disponible.
- La mémoire est divisée en deux parties M_1 et M_2 qui fonctionnent en "bascule". Pendant que *A* écrit dans M_1, *B* exploite le contenu de M_2 qui a été inscrite antérieurement. Quand M_1 est pleine, *A* remplit M_2 qui a été vidée par *B*. *B* traite alors le contenu de M_1. Cette méthode augmente la vitesse de traitement de la machine puisque l'on traite des informations alors qu'on en acquiert de nouvelles.
- La mémoire est organisée en file, et gérée par une logique de commande qui inscrit les nouvelles données avec une adresse d'écriture et qui lit les données les plus anciennes avec une adresse de lecture. L'adresse d'écriture est "en avance" sur l'adresse de lecture. Si l'adresse de lecture rattrape l'adresse d'écriture, la mémoire est vide. Si c'est l'adresse d'écriture qui rattrape l'adresse de lecture, la mémoire est pleine. Une logique de gestion transforme ainsi la RAM en une FIFO. Cette technique s'emploie surtout dans un système qui comporte une mémoire importante et dans laquelle une fraction de celle-ci sert d'interface à deux utilisateurs.
- La mémoire est organisée en file, comme précédemment, mais plutôt que d'utiliser une fraction de mémoire, on se sert d'une mémoire FIFO intégrée. Dans ce cas, il n'y a plus d'adresse à gérer. On se contente de tester si la mémoire est vide ou pleine. Les entrées et sorties sont alors réellement simultanées.

Pour que ces systèmes fonctionnent correctement, il faut:

- Que le rythme moyen de prélèvement soit supérieur au rythme moyen d'écriture dans les mémoires.
- Que la capacité des mémoires soit suffisante pour absorber une arrivée brusque de données.

Si ces conditions ne sont pas remplies temporairement, il faut pouvoir maîtriser le débit des données entrantes. Si ce n'est pas possible, des données se perdent.

La figure 16.3 donne des diagrammes d'évolution des mémoires tampons abordées dans ce paragraphe.

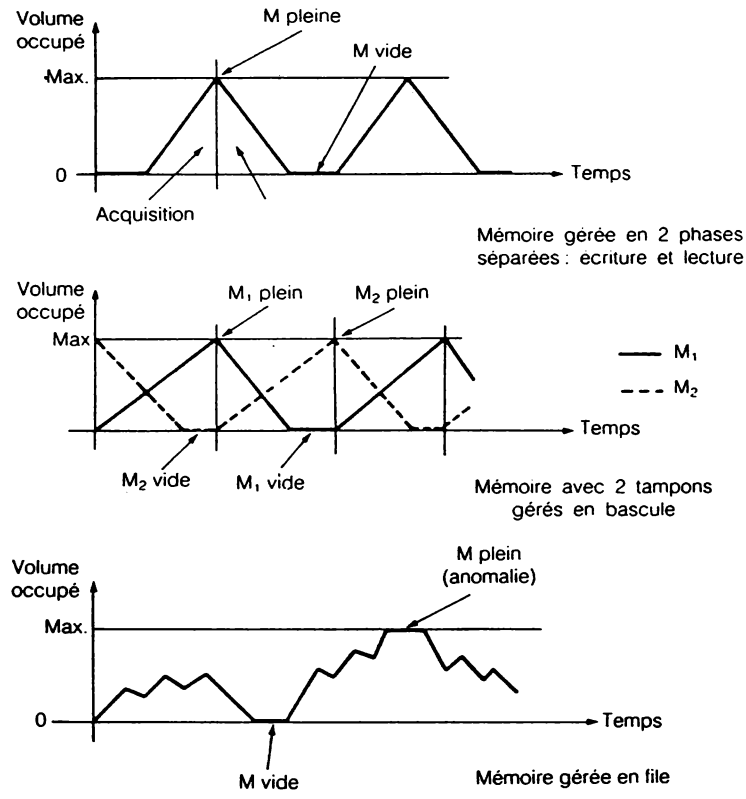


Fig. 16.3. — *Quelques exemples de gestion de mémoire.*

5. TRANSCODAGE

Nous avons déjà traité le transcodage par le calcul booléen au chapitre 2. On écrit chacun des éléments binaires du code d'arrivée en fonction des éléments du code de départ. Une solution plus simple, qui évite tout calcul, consiste à appliquer les éléments binaires du code de départ sur les entrées adresse d'une PROM. Dans chacun des mots correspondant au code de départ on inscrit la valeur du code d'arrivée (fig. 16.4).

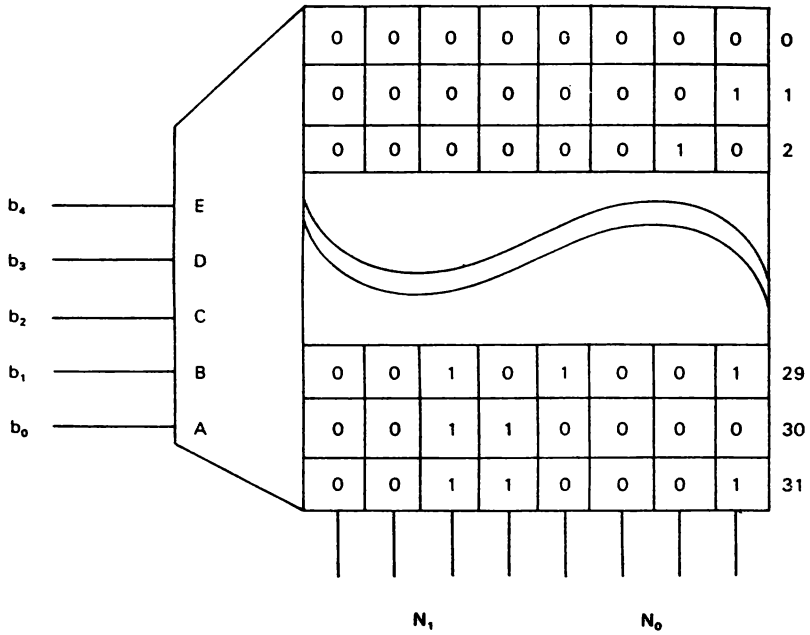


Fig. 16.4. — *Transcodage binaire → décimal codé binaire avec une PROM.*

6. MACHINE SÉQUENTIELLE SIMPLE

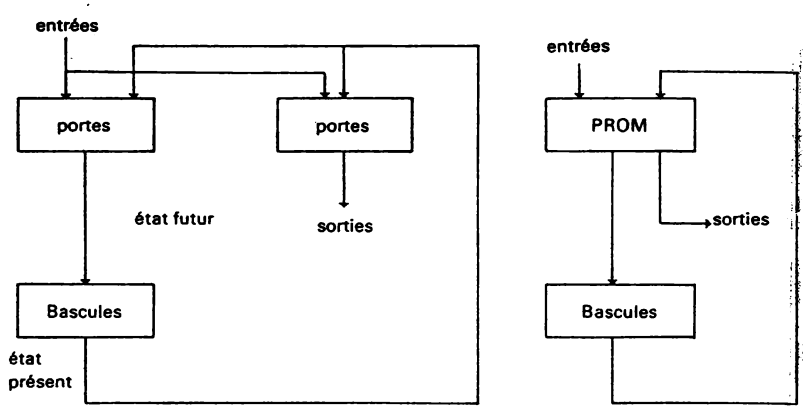


Fig. 16.5. — *Structure d'une machine séquentielle simple.*

Une machine séquentielle simple comprend une bloc mémoire, donnant l'état de la machine et deux blocs combinatoires, l'un calculant l'état futur du système et l'autre les sorties. Les portes peuvent être remplacées par une PROM.

A l'instant t_1, on inscrit dans le registre de mémorisation l'état des entrées et des bascules du système. Cette nouvelle configuration donne une nouvelle adresse, amenant en sortie de la mémoire un nouvel état des actions et le prochain état des bascules. Le temps t_2, disjoint de t_1 permet d'exploiter les eb pour les actions.

7. MÉMOIRE PROM UTILISÉE EN GÉNÉRATEUR DE FONCTIONS

Les deux exemples précédents, transcodage et système séquentiel, montre qu'une mémoire PROM reçoit des informations d'entrée sur ses adresses et fournit en réponse des informations sur ses sorties. On peut établir des relations booléennes entre les entrées et les sorties, de sorte qu'une mémoire morte peut remplacer des portes pour engendrer des fonctions booléennes. Nous avons vu dans le chapitre précédent qu'il existait aussi des circuits programmables pour ce type de fonction : les FPLA et les PAL.

8. RÉALISATION DES MÉMOIRES DANS UN SYSTÈME

8.1. Mémoires intégrées dans une logique de gestion

Dans le cas de petits systèmes, la matrice de boîtiers mémoires s'intègre directement à la logique de gestion. La figure 16.6 donne un exemple de structure. On trouve :

- La matrice de boîtiers réalisant la capacité et le format nécessaire (comme indiqué au paragraphe 18 du chapitre précédent).
- Un registre d'adresse RAM programmable (chargement, incrémentation, remise à zéro).

- Associé au registre RAM, un circuit DEC qui décode les poids forts de RAM pour les sélections dans la matrice de boîtiers. Les eb de poids faibles adressent directement les boîtiers.
- Un registre d'entrée-sortie RES qui reçoit soit une donnée venant de la mémoire soit une donnée extérieure à la mémoire, à travers un multiplexeur MUX. La sortie de RES est connectée à l'entrée information de la mémoire.
- Les signaux divers de gestion, comme *R/W* (lecture/écriture), les horloges, les signaux d'actions sur RAM et RES proviennent directement de la logique de gestion.

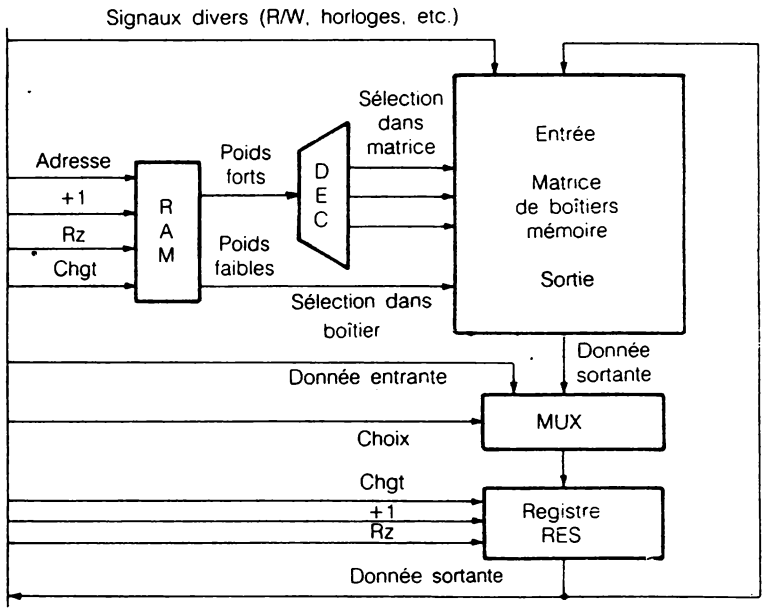


Fig. 16.6. — *Mémoire associée à une logique de gestion.*

8.2. Mémoires organisées en cartes indépendantes

8.2.1. Principe

Pour les grands systèmes de traitement d'information, les boîtiers mémoires sont organisés en sous-ensembles distincts et forment des cartes de mémoires, de capacité et de format donné: 16 *K* mots de 8 eb par exemple. Le système est alors modulaire. Les fonctions de traitement sont

localisées et les fonctions mémoires également. L'utilisateur juxtapose un certain nombre de cartes pour obtenir le volume nécessaire.

Les liaisons entre l'organe de traitement et les mémoires se simplifient. Il n'y a plus cette diversité de signaux comme pour une mémoire intégrée. Le dialogue s'effectue au moyen de *liaisons bus*. La gestion de l'adresse et des données n'est plus faite au voisinage de la matrice mémoire mais dans la logique de gestion. On trouve :

- Un bus d'adresse qui transporte tous les eb permettant l'adressage de tout l'espace mémoire gérable par la logique de gestion. Chaque carte mémoire représente une partie de cet espace adressable. *Exemple* : cartes mémoires de 16 *K* mots de 8 eb pour un espace adressable de 128 *K* mots de 8 eb.
- Un bus de donnée, bidirectionnel. Il émet ou reçoit sur les mêmes fils la donnée à écrire dans la mémoire ou la donnée lue dans la mémoire.
- Un bus de commande comprenant des signaux de gestion en petit nombre comme écriture/lecture, horloges, mémoire disponible, etc...

La figure 16.7 montre la connexion entre la logique de gestion et les cartes mémoires. Les trois bus alimentent toutes les cartes en même temps.

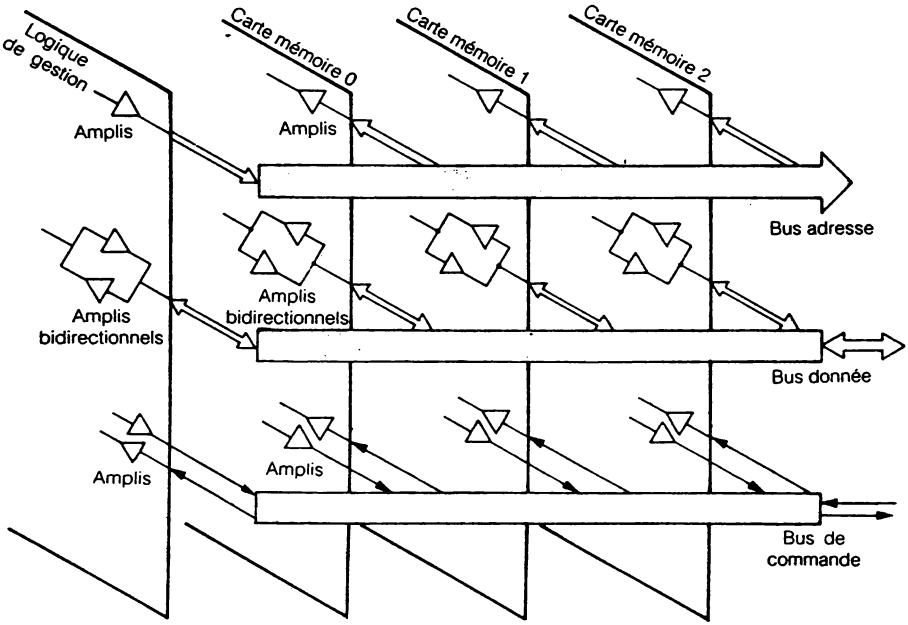


Fig. 16.7. - *Cartes mémoires associées à une logique de gestion.*

- *Bus adresse*: des amplificateurs (ou buffers), situés sur la carte de gestion fournissent les eb d'adresse aux cartes, et des amplificateurs, situés sur les cartes de mémoire permettant de ne présenter qu'une seule charge pour chaque carte mémoire vis à vis de la carte de gestion. Les amplificateurs de la carte mémoire pilotent ensuite les nombreux boîtiers internes à la carte.
- *Bus donnée*: des amplificateurs bidirectionnels (ou transceivers) situés sur la carte de gestion fournissent ou reçoivent les données du bus donnée. Des amplificateurs bidirectionnels assurent les fonctions symétriques sur les cartes mémoires. Le sens d'ouverture des "transceivers" est déterminé par le type d'opération effectuée (lecture ou écriture).
- *Bus commande*: idem bus adresse mais avec les deux sens possible.

8.2.2. Découpage de l'espace mémoire

L'espace mémoire est défini par plusieurs caractéristiques:

- Sa capacité totale, qui dépend du nombre d'eb utilisés pour adresser les mots. Avec 16 eb, on peut désigner $2^{16} = 65536$ mots distincts.
- Son format qui donne le nombre d'eb accessibles en même temps (taille du mot).
- Les zones effectivement utilisées dans l'espace mémoire. Si toutes les adresses sont utilisées, l'adressage est continu de la valeur 0 à $2^n - 1$. Si des zônes ne sont pas utilisées, l'adressage est discontinu.
- La façon dont les zones utilisées sont "pavées". Ce "pavage" dépend de la capacité et de la taille des boîtiers employés. Comme l'espace mémoire comprend de la mémoire morte, vive, statique ou dynamique, etc... le pavage n'est pas uniforme.

La figure 16.8 représente un espace mémoire de 64 *K* de 8 eb avec un adressage discontinu: les zones utilisées vont de 0 à 12 287 (0 à 12 *K*), de 36 864 à 45 055 (36 *K* à 44 *K*) et de 53 248 à 65 535 (52 *K* à 64 *K*), par tranche de 4 *K*.

La figure 16.9 montre un pavage non uniforme de l'espace mémoire discontinu de la figure 16.8. La tranche T_{12}, bien que physiquement présente à cause de la modularité 16 *K* × 1 des boîtiers de mémoire dynamique n'est pas utilisée.

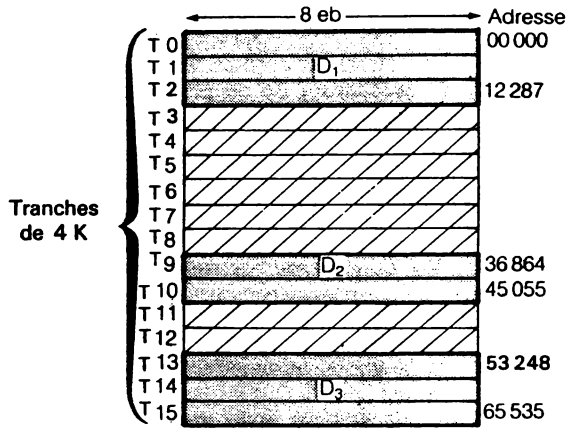


Fig. 16.8. — *Espace mémoire du 64 K de 8 eb à adressage discontinu.*

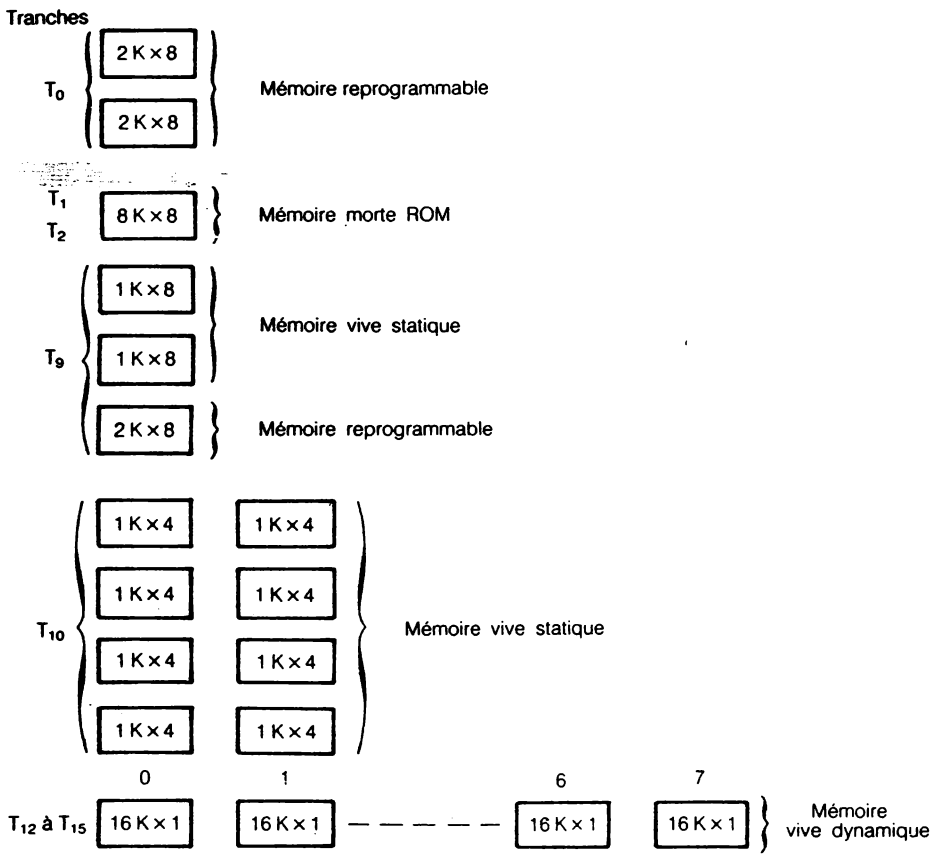


Fig. 16.9. — *Pavage non uniforme de l'espace mémoire discontinu.*

La cartographie ou "mapping" d'un espace mémoire décrit sa constitution et son pavage. Le "mapping" détermine les fonctions à calculer à partir des eb d'adresse pour valider les cartes, les rangées de boîtiers dans les cartes et les mots dans les boîtiers.

Chapitre 17

ASSEMBLAGE DE BOITIERS MÉMOIRE

1. INTRODUCTION

Comme il a été vu au chapitre 15, il existe de nombreux types de mémoire : vive statique et dynamique, ROM, EPROM, EEPROM, FIFO. Décrire ces boîtiers par le détail serait fastidieux et trop long. Aussi avons-nous choixi de ne traiter qu'un seul boîtier, le TMS 2114, mémoire vive statique de 1 024 mots de 4 eb. Ce boîtier sera utilisé ensuite pour concevoir une carte mémoire.

2. BROCHAGE DU BOITIER

Les 18 broches du boîtier (fig. 17.1) se répartissent en :

- deux pour l'alimentation V_{cc} (+ 5 V) et la masse,
- quatre pour les quatre entrées-sorties (DQ_1^4),
- une pour la sélection du boîtier ($\overline{S}$),
- une pour l'écriture ($\overline{W}$),
- dix pour l'adresse A_0^9 ($2^{10} = 1\,024$).

On constate que pour économiser le nombre de broches et réduire l'encombrement, les broches d'entrées et sorties sont confondues.

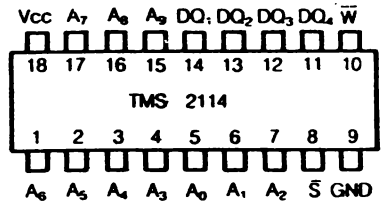


Fig. 17.1. — *Brochage du boîtier TMS 2114.*

3. PRINCIPE D'EXPLOITATION

Le fonctionnement du boîtier est déterminé essentiellement par les deux entrées $\overline{S}$ et $\overline{W}$.

$\overline{S} = 1$: le boîtier est "hors service". La commande $\overline{W}$ n'a pas d'effet, ni en lecture, ni en écriture. Les sorties DQ sont en mode haute impédance.

$\overline{S} = 0$ et $\overline{W} = 0$: les buffers de sortie (amplificateurs) sont en mode haute impédance. L'information appliquée sur les broches DQ est écrite dans la cellule.

$\overline{S} = 0$ et $\overline{W} = 1$: les buffers de sortie sont validés et l'état de la cellule choisie apparaît sur les broches DQ.

Le point délicat dans l'exploitation de cette mémoire à entrées-sorties communes est de n'avoir jamais simultanément une information extérieure et l'état d'une cellule mémoire présents sur les broches d'entrée-sortie. Auquel cas on dit qu'il y a un conflit de bus, très génant car on risque la destruction du boîtier si ce conflit dure trop longtemps ou un fonctionnement erratique pour un conflit momentané.

Le détail du fonctionnement est donné dans les paragraphes suivants, qui montrent le diagramme des temps des opérations de lecture et d'écriture avec les contraintes temporelles à respecter.

4. CYCLE DE LECTURE

C'est le fonctionnement le plus simple (fig. 17.2). La procédure consiste à :

- établir l'adresse de la cellule choisie, $\overline{W}$ étant déjà à 1,
- établir le chip select ($\overline{S}$) à l'état 0 puis à l'état 1,
- l'information apparaît en sortie puis disparaît quand chip select revient à l'état 1.

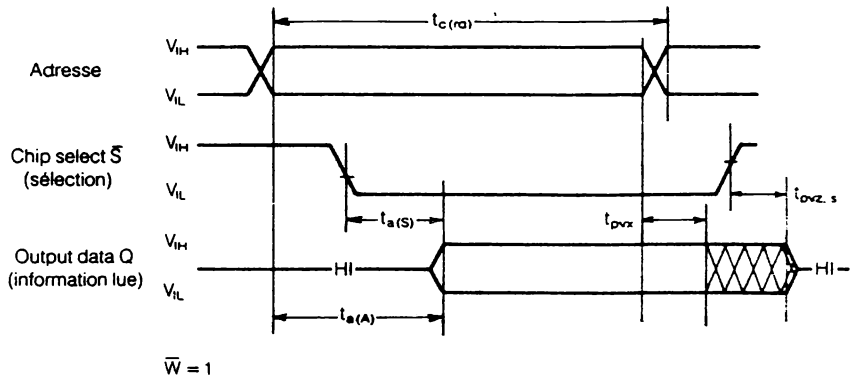


Fig. 17.2. – *Diagramme d'un cycle de lecture.*

Détaillons les paramètres associés à ce cycle de lecture :

- $t_{c(rd)}$: durée du cycle de lecture,
- $t_{a(A)}$: durée du temps d'accès par rapport à l'établissement de l'adresse,
- $t_{a(S)}$: durée du temps d'accès par rapport au front de descente de $\overline{S}$,
- t_{PVX} : durée de conservation de l'information après un changement d'adresse,
- $t_{PVZ,S}$: temps de désélection de la sortie et retour au mode haute impédance par rapport au front de montée de $\overline{S}$.

On pourrait également obtenir un cycle de lecture plus simple en maintenant constants chip select à 0 et write enable à 1 (fig. 17.3) ; les sorties sont seulement sous la dépendance de l'adresse.

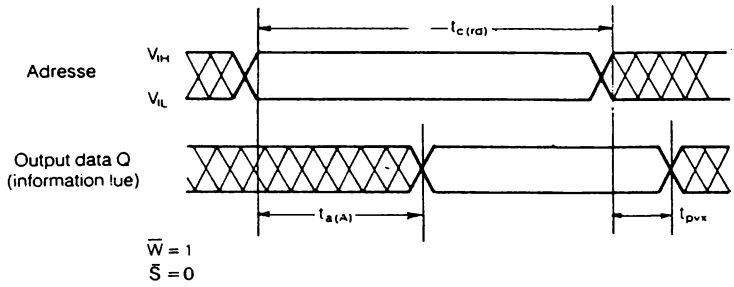


Fig. 17.3. — *Cycle de lecture avec* $\overline{S} = 0$.

Les paramètres $t_{a(A)}$ et t_{PVX} des figures 17.2 et 17.3 se rapportent tous deux au changement d'adresse et au changement de donnée. Ils sont cependant différents. Le premier $t_{a(A)}$ exprime au bout de combien de temps l'information est stabilisée et peut être lue tandis que le second exprime combien de temps l'information de la cellule sélectionnée reste stable avant de disparaître. Le premier paramètre donne l'instant de lecture au plus tôt et le second paramètre l'instant de lecture au plus tard.

Notons que ce cycle avec $\overline{S} = 0$ maintenu de manière constante n'est applicable que dans le cas où les boîtiers sont isolés et non connectés par leur sortie (fig. 17.4).

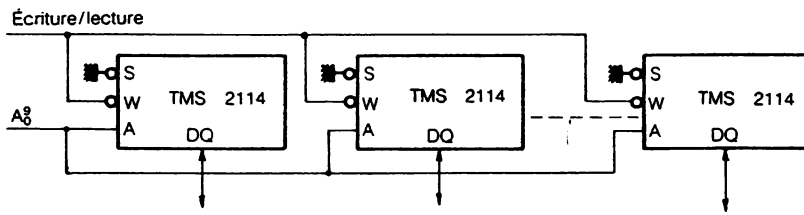


Fig. 17.4. — *Exploitation des boîtiers TMS 2114 avec S constant.*

5. CYCLE D'ÉCRITURE AVANCÉ (EARLY WRITE CYCLE)

Il est plus compliqué que le précédent (fig. 17.5). Il faut :

— établir l'adresse,

– envoyer une impulsion d'écriture ($\overline{W} = 0$),

– envoyer une impulsion de sélection ($\overline{S} = 0$) décalée par rapport à celle d'ecriture.

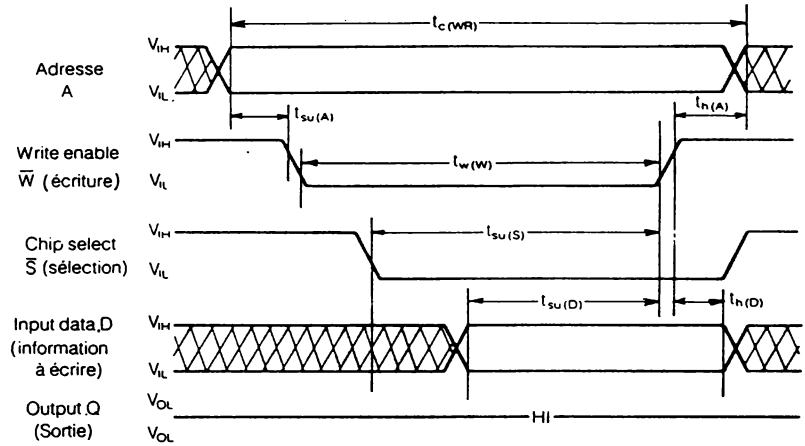


Fig. 17.5. – *Cycle d'écriture avancé (early write cycle).*

– Présenter l'information sur l'entrée D un certain temps avant et après le front de montée de $\overline{W}$.

Le cycle de la figure 17.5 est dit à écriture avancée (early write cycle) parce que l'impulsion d'écriture $\overline{W}$ apparaît avant celle de sélection ($\overline{S}$). Dans ce cas, la sortie Q reste à l'état haute impédance pendant tout le cycle, ce qui permet d'afficher une donnée sur l'entrée D (commune à la sortie Q) sans avoir de conflit de bus. A la fin du cycle d'écriture quand $\overline{W}$ est revenu à 1 et que $\overline{S}$ vaut encore 0, on se trouve normalement dans les conditions statiques d'un cycle de lecture ($\overline{W} = 1$, $\overline{S} = 0$). La sortie Q qui devrait afficher alors le contenu de la case mémoire reste cependant à l'état haute impédance. Ceci montre que le fonctionnement du boîtier dépend aussi de l'ordre dans lequel les signaux ont été appliqués. Les paramètres associés à ce cycle d'écriture sont :

– $t_{c(Wr)}$: durée du cycle d'écriture,

– $t_{su(A)}$: temps de préaffichage de l'adresse par rapport au front de descente de $\overline{W}$,

– $t_{h(A)}$: temps de maintien de l'adresse par rapport au front de montée de $\overline{W}$,

– $t_{w(W)}$: durée du signal d'écriture $\overline{W}$.

— $t_{su(S)}$: temps de préaffichage de $\overline{S}$ par rapport au front de montée de $\overline{W}$,

— $t_{su(D)}$: temps de préaffichage de l'information entrante par rapport au front de montée de $\overline{W}$,

— $t_{h(D)}$: temps de maintien de l'information entrante par rapport au front de montée de $\overline{W}$.

6. CYCLE DE LECTURE-MODIFICATION-ÉCRITURE

Dans une machine de traitement d'information on est amené à considérer parfois comme une seule opération la lecture d'une case mémoire, la modification de la valeur lue, et la réinscription de la nouvelle valeur à la même adresse. Le cycle est alors appelé lecture-modification-écriture. La figure 17.6 montre un tel cycle. La procédure à suivre consiste à :

— établir l'adresse, $\overline{W}$ et $\overline{S}$ étant à 1,

— envoyer une impulsion $\overline{S}$ à 0,

— envoyer à l'intérieur de l'impulsion $\overline{S}$, une impulsion $\overline{W}$ à 0,

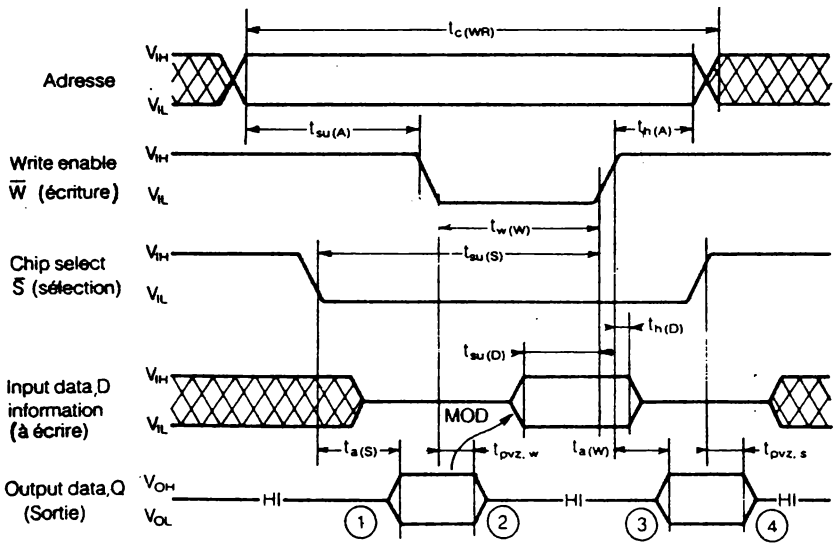


Fig. 17.6. — *Cycle de lecture, modification, écriture.*

– présenter l'information à écrire autour du front montant de $\overline{W}$. Cette information résulte de celle obtenue en lecture dans la première phase du cycle dans $\overline{W} = 1$ et $\overline{S} = 0$ après modification par des circuits de calcul.

Les paramètres nouveaux dans ce cycle sont :

– $t_{PVZ,W}$: temps de désélection de la sortie après Write Enable bas,

– $t_{a(W)}$: temps d'accès depuis Write Enable haut.

Les indications ①, ②, ③, ④ portées à la figure 17.6 correspondent à des conflits de bus éventuels. Il y a risque de recouvrement entre les informations venant de l'extérieur et celles sortant de la mémoire.

7. TABLEAU DES PARAMÈTRES

Les figures 17.7, 17.8, 17.9 donnent les paramètres électriques et temporels du boîtier TMS 2114-15 (150 *ns* de temps accès), le plus rapide d'une famille qui comprend les boîtiers TSM 2114-(15-20-25-45) de temps d'accès respectifs 150, 200, 250 et 450 *ns*.

Paramètre		MIN	NOM	MAX	Unité
Tension d'alimentation : Vcc	En fonctionnement	4,5	5	5,5	V
	A puissance réduite	2,4	5	5,5	
Tension d'alimentation : Vss			0		V
V_{IH} : Tension d'entrée niveau haut		2,0		5,5	V
V_{IL} : Tension d'entrée niveau bas		−1,0		0,8	V
T_A : Température de fonctionnement		0		70	°C

Fig. 17.7. – *Paramètres électriques.*

Paramètre	Condition de test		MIN	TYP	MAX	Unité
V_{OH} : Tension de sortie niveau haut	$I_{OH} = -1,0$ mA Vcc = 4,5 V		2,4			V
V_{OL} : Tension de sortie niveau bas	$I_{OL} = 3,2$ mA Vcc = 4,5 V				0,4	V
I_I : Courant d'entrée	$V_I = 0$ V à 5,5 V				10	µA
I_{OZ} : Courant de sortie à l'état haute impédance	$\overline{S}$ à 2 V ou $\overline{W}$ à 0,3 V $V_O = 0$ V à 5,5 V		-10		10	µA
I_{CC} Courant d'alimentation	$I_O = 0$ mA	Vcc = 5,5 V		90	100	mA
	$T_A = 0$ °C	Vcc = 2,4 V		60	70	mA
C_I : capacité d'entrée	$V_I = 0$V f = 1 MHz				8	pF
C_O : capacité de sortie	$V_O = 0$V f = 1 MHz				8	pF

Fig. 17.8. — *Paramètres électriques.*

Paramètres	MIN	MAX	Unité
$t_{c(rd)}$: temps de cycle de lecture	150		ns
$t_{c(WR)}$: temps de cycle d'écriture	150		ns
$t_{w(W)}$: durée de l'impulsion d'écriture	80		ns
$t_{su(A)}$: temps de préaffichage de l'adresse	0		ns
$t_{su(S)}$: temps de préaffichage du chip select	80		ns
$t_{su(D)}$: temps de préaffichage de l'information	80		ns
$t_{h(D)}$: temps de maintien de l'information	0		ns
$t_{h(A)}$: temps de maintien de l'adresse	0		ns
$t_{a(A)}$: temps d'accès depuis l'adresse		150	ns
$t_{a(S)}$: temps d'accès depuis le chip select		70	ns
$t_{a(W)}$: temps d'accès depuis write enable haut		70	ns
t_{pvz} : temps de conservation de l'information après changement d'adresse	20		ns
$t_{pvz.s}$: temps de désélection de la sortie après chip select haut		50	ns
$t_{pvz.w}$: temps de désélection de la sortie après write enable bas		50	ns

Conditions : t_A : 0 °C à 70 °C, toute la gamme de tension recommandée.
1 charge TTL, C_L = 100 pF

Fig. 17.9. — *Paramètres électriques.*

8. SPÉCIFICATION DE LA CARTE MÉMOIRE

On étudie dans ce qui suit l'organisation d'une carte mémoire de capacité 16 K mots de 8 eb permettant de couvrir un espace adressable de 64 K mots de 8 eb (4 cartes sont nécessaires). La liaison entre la carte de gestion et les cartes mémoires comprend :

- un bus d'adresse de 16 eb A_0^{15} ($2^{16} = 65536$),
- un bus bidirectionnel de donnée à 8 eb BD_0^7,
- un signal de sélection S,
- un signal d'écriture W,
- un signal DR indiquant le sens du transfert sur le bus donnée : $DR = 0$ correspond à une écriture dans les mémoires et $DR = 1$ à une lecture des mémoires,
- les signaux entrant sur une carte mémoire ne doivent compter que pour une seule charge. Ils ont donc tous un amplificateur séparateur en entrée. De même tous les signaux sortant de la logique de gestion sont fournis à travers un amplificateur séparateur.

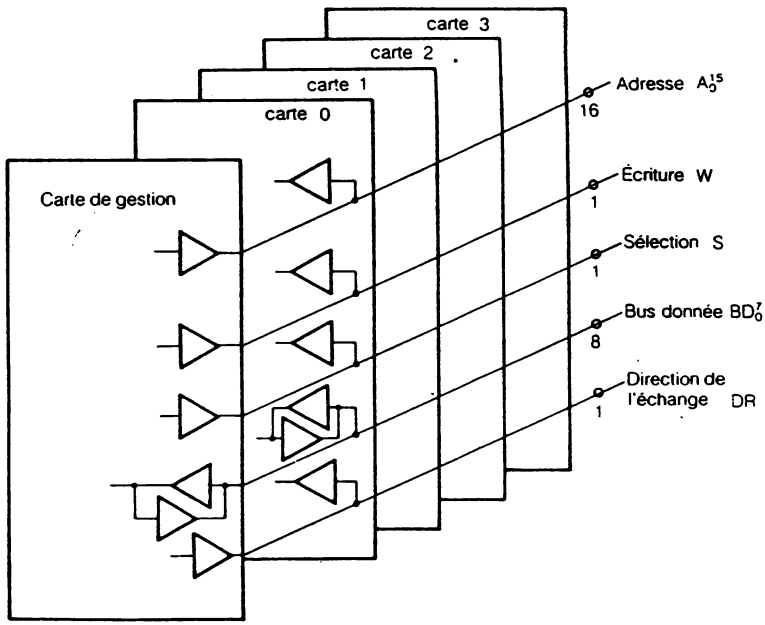


Fig. 17.10. — *Interface entre logique de gestion et cartes mémoires.*

On utilise le boîtier mémoire TMS 2114 décrit dans les paragraphes précédents, dont la figure 17.11 donne le schéma logique.

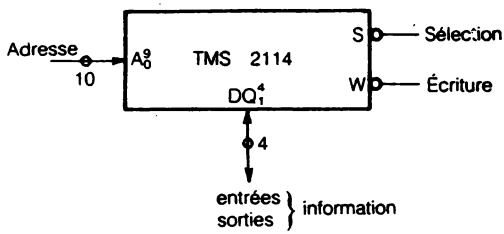


Fig. 17.11. — *Boîtier mémoire TMS 2114.*

Une opération d'écriture nécessite une adresse, une information sur DQ, un signal d'écriture W à 0 et de sélection S à 0. Une opération de lecture demande une adresse, un signal d'écriture W à 1 et un signal de sélection S à 0, l'information lue apparaissant sur la sortie DQ.

9. CONSTITUTION DE LA MATRICE MÉMOIRE

Une carte mémoire de 16 K mots comprend une matrice de 32 boîtiers répartis en 16 lignes de deux boîtiers, chaque ligne représentant 1 K mot de 8 eb. L'adresse globale A_0^{15} d'un mot quelconque doit permettre de désigner :

- la carte parmi quatre dans laquelle se trouve le mot,
- dans la carte choisie, le numéro de la ligne,
- dans la ligne choisie, le numéro dans le boîtier.

L'adresse A_0^{15} peut se décomposer ainsi :

$$(A_0^{15} = am) = (A_{14}^{15} = ac).(A_{10}^{13} = al).(A_0^9 = ab)$$

↓	↓	↓	↓
adresse du mot	adresse de la carte	adresse de la ligne	adresse dans le boîtier

Le premier terme $A_{14}^{15} = ac$ correspond à la reconnaissance d'adresse de la carte. S'il y a égalité, une carte parmi quatre devient active, les trois autres restant inactives. La figure 17.12 montre différentes solutions pour réaliser cette reconnaissance d'adresse : comparateur d'adresse, décodeur,

mémoire PROM. La solution avec PROM évite tout câblage mais chaque carte à sa propre PROM.

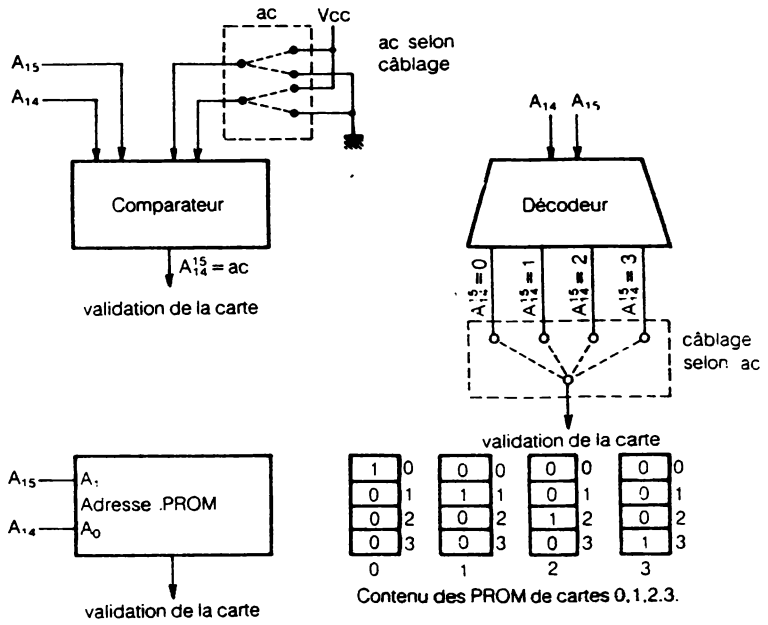


Fig. 17.12. — *Différentes méthodes de reconnaissance d'adresse.*

Le second terme $A_{10}^{13} = al$, ou reconnaissance de ligne s'obtient par décodage. Ce décodage est validé par la reconnaissance d'adresse de carte et par le signal de sélection. De la sorte seule une ligne de la carte sélectionnée est activée.

Le troisième terme $A_0^9 = ab$ s'obtient en mettant simplement l'adresse A_0^9 sur les entrées adresse du boîtier.

Le signal $\overline{W}$ est câblé en commun pour tous les boîtiers d'une carte. Il ne sera actif que pour la ligne sélectionnée.

L'accès au bus de donnée se réalise par un circuit émetteur/récepteur commandé par le signal *DR*.

La figure 17.13 donne la structure de la carte 16 $K \times 8$ correspondant à cette analyse. On y trouve :

— des boîtiers amplificateurs de type 74 LS 744 pour les signaux A_0^{15}, $\overline{W}$, *S*, *DR*,

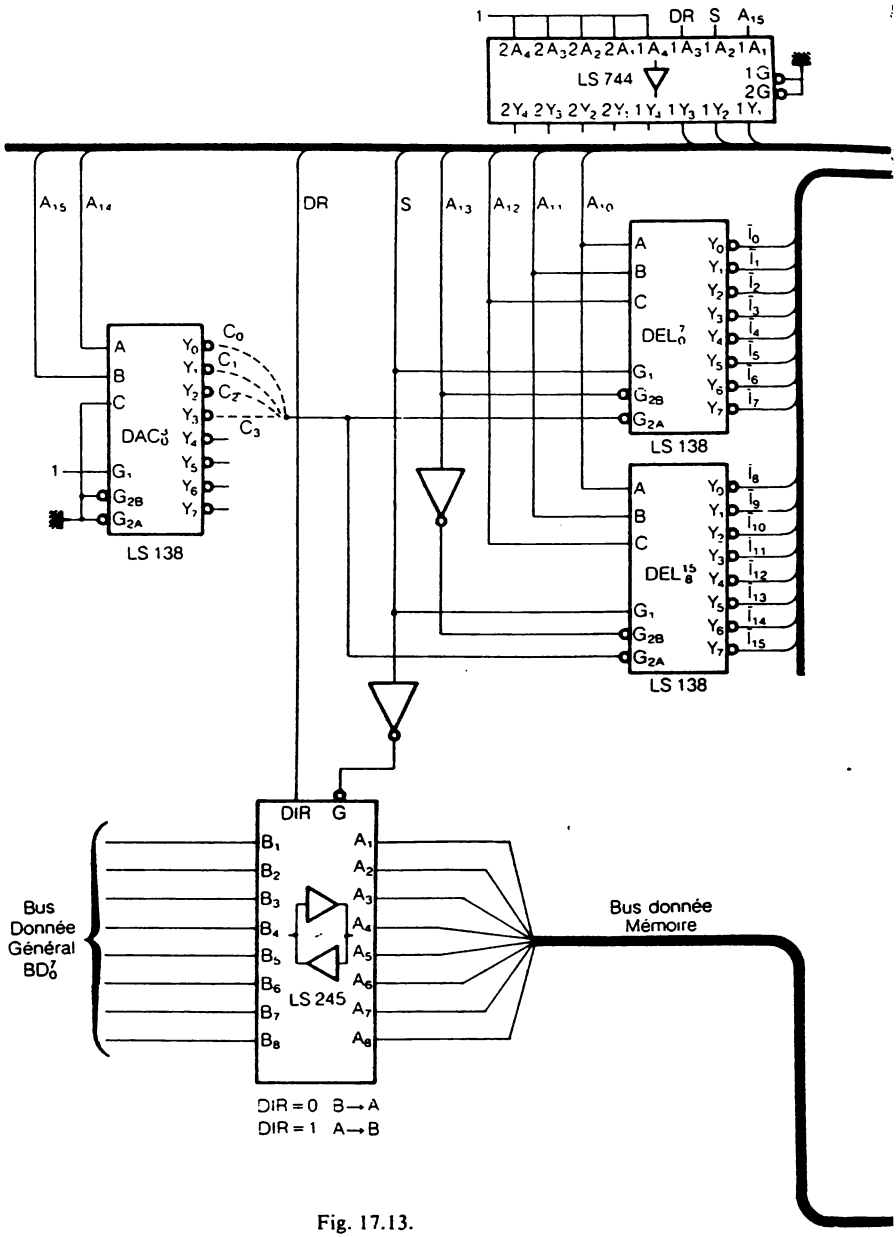
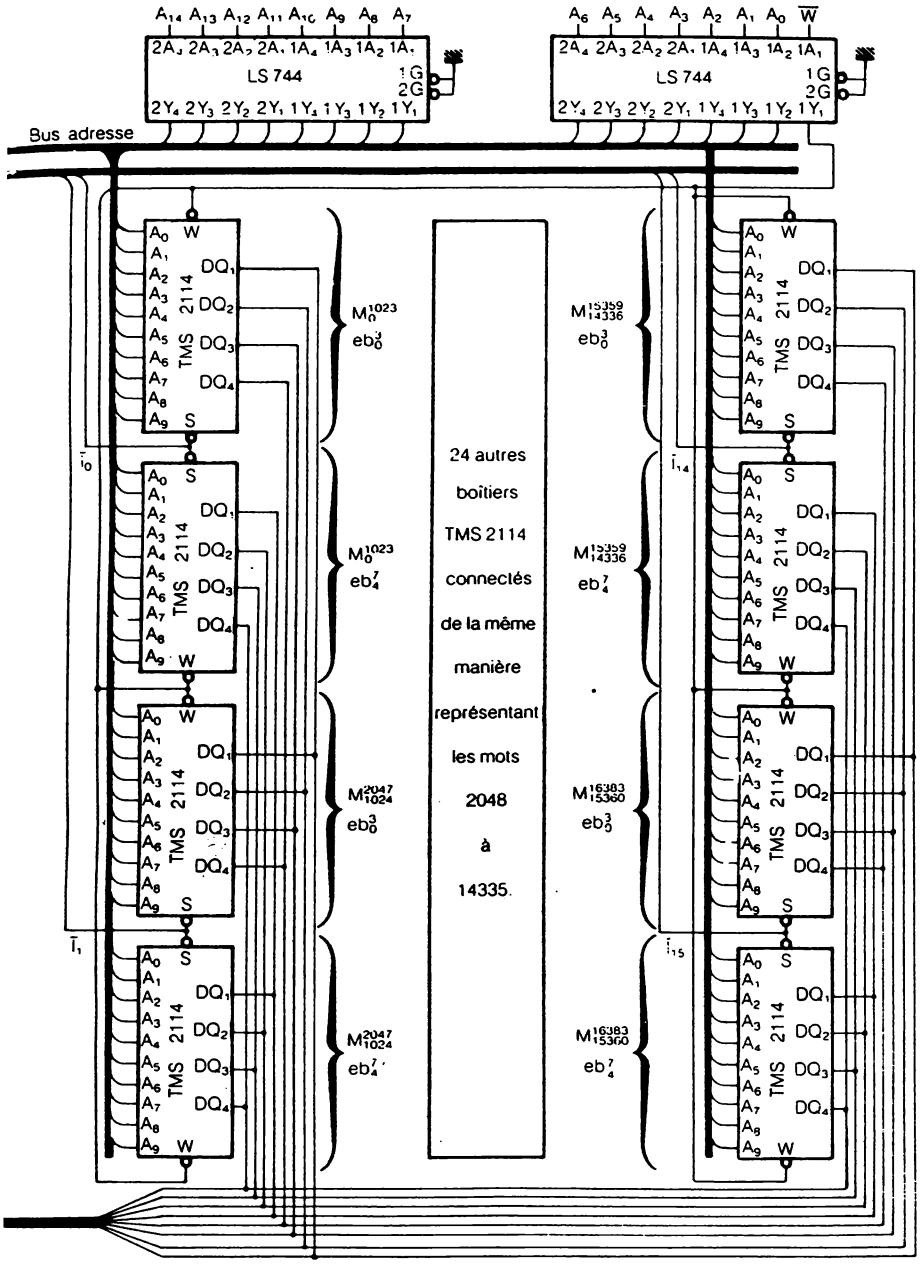

1
DR S A15
2A4 2A3 2A2 2A1 1A4 1A3 1A2 1A1
LS 744
1G
2G
2Y4 2Y3 2Y2 2Y1 1Y4 1Y3 1Y2 1Y1
A15 A14 DR S A13 A12 A11 A10
DAC0 7
LS 138
DEL0 7
LS 138
DEL8 15
LS 138
DIR G
LS 245
Bus Donnée Général BD0 7
Bus donnée Mémoire
DIR = 0 B → A
DIR = 1 A → B

Fig. 17.13.


LS 744
LS 744
Bus adresse
TMS 2114
24 autres boîtiers TMS 2114 connectés de la même manière représentant les mots 2048 à 14335.

- un décodeur d'adresse carte DAC_0^3 de type 74 LS 138 pour la reconnaissance du numéro de carte sur les eb A_{14}^{15},
- un décodeur d'adresse de ligne DEL_0^{15} réalisé avec deux décodeurs 74 LS 138, à partir des eb A_{10}^{13}, validé par la reconnaissance de carte et par le signal S,
- la matrice de 32 boîtiers de mémoire. Les boîtiers sont groupés par deux pour former un mot de 8 eb. Il y a seize groupes validés chacun par la sortie correspondante du décodeur de ligne DEL sur l'entrée S. Les boîtiers reçoivent tous les 10 eb d'adresse A_0^9 et le signal $\overline{W}$. Les entrées-sorties DQ sont reliées ensemble pour former le Bus Donnée Mémoire,
- un circuit émetteur-récepteur 74 LS 245 qui assure l'échange des données sur le bus donnée dans les deux sens, grâce au signal DR qui définit le sens de l'échange et au signal S qui valide le transfert.

10. RÉSOLUTION DES CONFLITS DE BUS

La mise en commun des entrées et sorties permet de réduire le nombre de broches, mais il faut veiller à éviter les conflits entre les signaux d'entrée et de sortie.

La figure 17.14 présente le schéma bloc général d'une carte 16 $K \times 8$ avec :

- décodage de A_{14}^{15} pour valider la carte (DAC_0^3),
- décodage de A_{10}^{13}, validé par la sortie de DAC_0^3 pour sélectionner un couple de boîtiers,
- réunion des entrées-sorties pour former le bus mémoire,
- validation du bus donnée par le signal de sélection S dans le circuit émetteur-récepteur.

Ce schéma ne diffère de celui de la figure 17.13 que par l'absence de validation par S au niveau du décodeur DEL. Au niveau des principes il paraît aussi bon que le précédent puisqu'un seul couple de boîtiers se trouve sélectionné à l'issue du décodage.

Si l'on applique un cycle d'écriture avancée (voir § 5, fig. 17.5), les sorties des boîtiers restent dans le mode trois états, puisque tous les boîtiers reçoivent $S = 1$ et $\overline{W} = 0$. Il n'y a donc pas de conflit entre les sorties des mémoires.

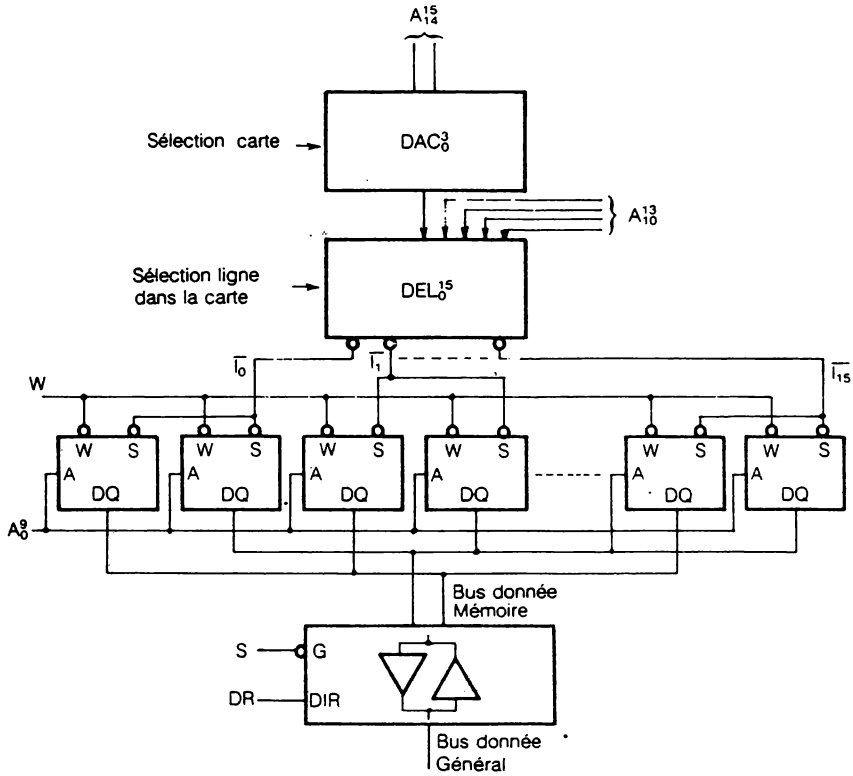


Fig. 17.14.

Pour un cycle de lecture maintenant, le signal $\overline{W}$ reste en 1. Les boîtiers sont sélectionnés à travers la chaîne de décodage comportant les amplificateurs d'entrée (74 LS 344), la PROM DAC_0^3, le décodeur DEL_0^{15}.

Cette chaîne complexe fournit des décodages transitoires au moment de la variation des adresses. Compte tenu des temps de réaction différents des boîtiers mémoires, des sorties vont se trouver simultanément en mode basse impédance, d'où des conflits temporaires de bus (fig. 17.15).

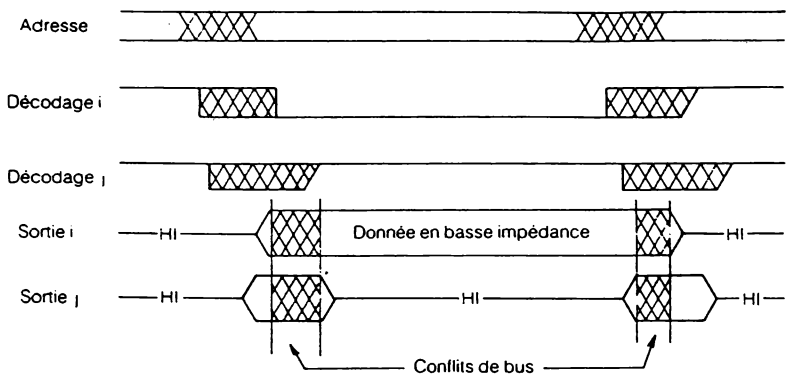


Fig. 17.15.

Le schéma de la figure 17.13, où l'on valide le décodeur DEL par *S*, supposé situé en dehors des zones de variation de l'adresse, on obtient une sélection seulement pour le couple de boîtiers sélectionné, les états transitoires étant ignorés (fig. 17.16).

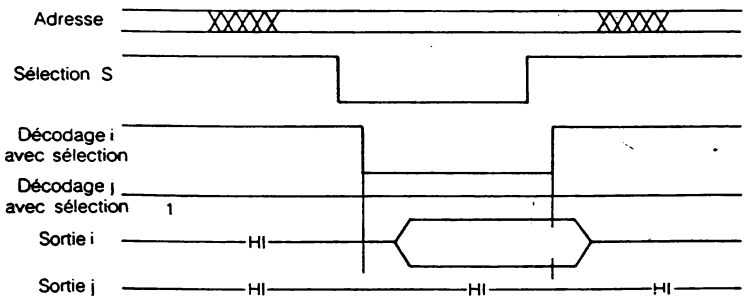


Fig. 17.16.

Chapitre 18

CONCEPTION DES SYSTÈMES SÉQUENTIELS SIMPLES

1. MÉTHODE DE CONCEPTION

Comme pour les systèmes combinatoires exposés au chapitre 9, la conception a pour but d'obtenir un schéma logique à partir des spécifications du produit. La figure 18.1 propose plusieurs façons pour y parvenir.

- L'analyse met en évidence un système simple, c'est-à-dire ayant peu d'entrées, peu de sorties, ou une structure itérative, composé de fonctions séquentielles connues. On réalise alors l'association de ces fonctions séquentielles dans un schéma-bloc ; on choisit ensuite les composants MSI ou LSI correspondants et l'on obtient le schéma.
- L'analyse met en évidence un système simple mais dans lequel n'apparaissent plus directement des fonctions séquentielles connues. On recense alors les variables d'entrée, les variables de sorties, les variables internes et l'on établit *un tableau de vérité* qui décrit l'évolution des variables internes et des sorties. On suppose que le montage est implicitement piloté par une ou plusieurs horloges. L'analyse par tableau de vérité, qui présente l'avantage d'être complet (on envisage tous les cas), ne met cependant pas en évidence certaines particularités temporelles. On peut alors compléter l'analyse par un *diagramme des temps*. On choisit ensuite les types de composants pour réaliser le schéma.

— Enfin l'analyse met en évidence un système plus complexe, dont le nombre d'entrées, sorties, variables internes est trop grand pour permettre l'analyse par un tableau de vérité. On décompose alors le système en plusieurs sous-systèmes séquentiels ou combinatoires composés de fonctions connues ou quelconques que l'on associe dans un schéma global. Aucune méthode générale n'existe pour effectuer ce découpage ; le concepteur doit faire appel à son expérience, à sa connaissance des fonctions intégrées et à son intuition. Pour décrire ou faire la synthèse d'un tel système on utilise des tableaux de vérité (mais seulement pour des sous-ensembles, le système entier étant trop complexe), des graphes, des organigrammes, des équations logiques, des schémas-blocs, des programmes. Le but est de déterminer les fonctions d'excitations des fonctions principales obtenues dans la phase de décomposition fonctionnelle du système et de réaliser ensuite ces fonctions d'excitation par des portes ou des circuits MSI, LSI. L'étude de ces systèmes complexes ne relève pas de cet ouvrage. Le lecteur pourra consulter "De la logique câblée aux microprocesseurs" tomes 3 et 4 aux éditions Eyrolles, par les mêmes auteurs.

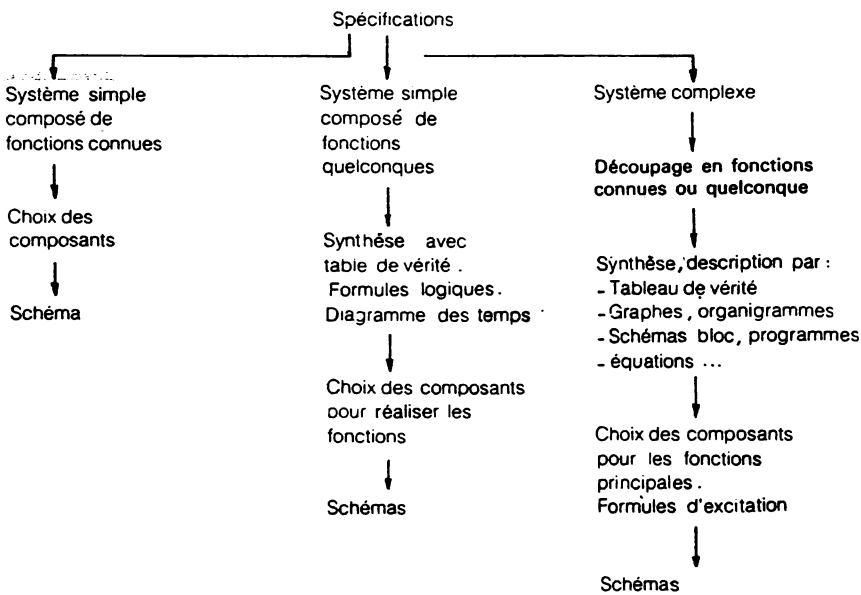


Fig. 18.1. — *Méthodes de synthèse.*

Afin d'illustrer la synthèse d'un système logique séquentiel simple nous étudierons trois cas :

- un système simple composé de fonctions connues : explorateur de voies,
- deux systèmes simples composés de fonctions quelconques : un détecteur synchrone de front de montée et de descente et un générateur de caractères.

2. SYSTÈME SIMPLE COMPOSÉ DE FONCTIONS CONNUES

Reprenons l'exemple traité au chapitre 9 pour un système combinatoire simple composé de fonctions connues où nous avons visualisé sur une lampe l'état ouvert ou fermé d'un dispositif mécanique parmi 64, en agissant par une commande manuelle sur les entrées d'adresse des multiplexeurs (fig. 9.3 et 9.4 du chapitre 9). Supposons maintenant que l'état de sortie du multiplexeur MUX_8 ne soit plus exploité visuellement mais traité par un système logique qui explore avec une période T les 64 dispositifs mécaniques, et intéressons-nous à la commande des multiplexeurs.

Il est évident qu'il suffit de réaliser un compteur à six étages commandé par une horloge de période $T/64$. Un tel compteur se réalise sans difficulté en utilisant des compteurs intégrés 74 LS 163 en cascade (fig. 18.2). Les six étages de poids faibles sont utilisés pour adresser les multiplexeurs et les étages de poids forts commutent mais ne sont pas utilisés.

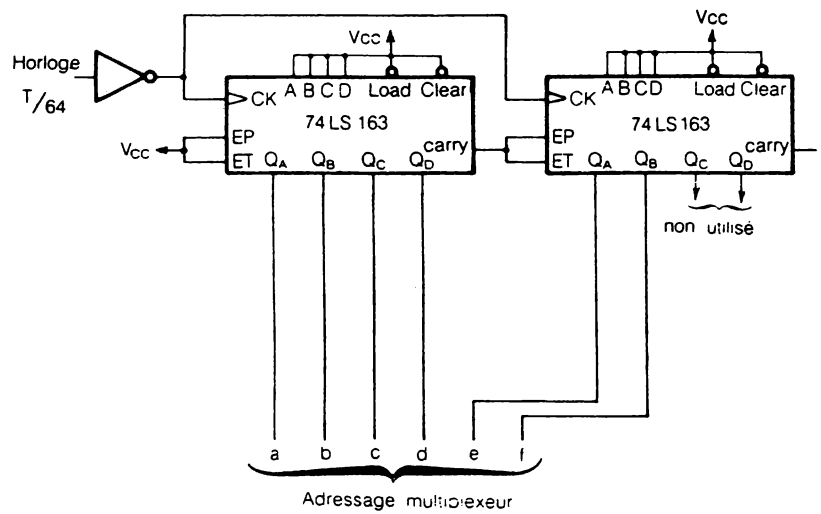


Fig. 18.2.

3. SYSTÈME SIMPLE COMPOSÉ DE FONCTIONS QUELCONQUES : DÉTECTION SYNCHRONE DE FRONTS

3.1. Synthèse avec tableau de vérité

Soit une impulsion Z dont on cherche à détecter le front de montée et le front de descente en créant deux fonctions distinctes : début de signal DS et fin de signal FS (fig. 18.3). A première vue l'énoncé de ce problème ne suggère pas de fonction séquentielle connue intégrée dans un boîtier. Il faut donc en réaliser l'étude. Supposons que nous disposions d'une horloge régulière h de période très inférieure à la durée de Z, que nous utilisons pour scruter l'état du signal Z (fig. 18.4).

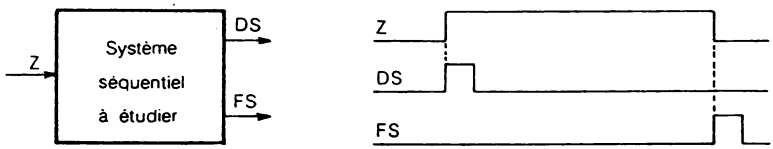


Fig. 18.3.

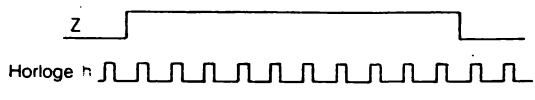


Fig. 18.4.

Le signal DS doit apparaître lorsque l'horloge, après avoir détecté un état $Z = 0$, détecte ensuite un état $Z = 1$ (passage de 0 vers 1) et le signal FS doit apparaître lorsque l'horloge détecte un état $Z = 0$ après avoir détecté un état $Z = 1$ (passage de 1 vers 0). Pour calculer de telles fonctions il est donc nécessaire de connaître au moment du test du signal Z son état antérieur. D'où l'idée d'introduire une variable interne qui mémorise l'état ancien de Z, soit Za. On établit la table de vérité de la figure 18.5 où apparaissent la variable externe Z, la variable interne Za, l'état futur de Za soit Za^+ et les deux fonctions de sortie DS et FS. On obtient les formules de DS et FS à partir desquelles on dessine le schéma de la figure 18.6.

Za	Z	Za^+	FS	DS
0	0	0	0	0
0	1	1	0	1
1	0	0	1	0
1	1	1	0	0

$Za^+ = Z$

$FS = Za\bar{Z}$

$DS = \bar{Z}aZ$

Fig. 18.5.

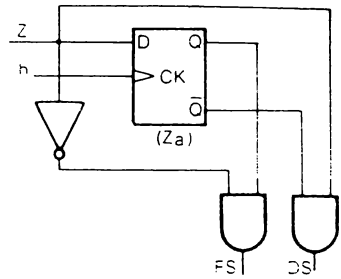


Fig. 18.6.

3.2. Diagramme des temps du montage

Le diagramme de la figure 18.7 donne le fonctionnement du circuit ainsi réalisé.

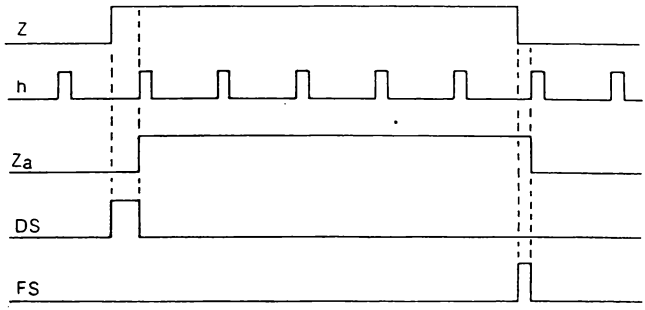


Fig. 18.7.

Ce diagramme montre que la durée des signaux *DS* et *FS* n'est pas constante et varie de zéro à la période *T* de l'horloge *h*. Le fonctionnement du circuit n'est donc pas correct. Cette particularité n'avait pas été mise en évidence par la table de vérité. L'analyse que nous venons d'effectuer montre la complémentarité qui existe entre table de vérité et diagramme des temps pour l'étude d'un petit système séquentiel.

Pour obtenir un fonctionnement correct il faut introduire une bascule de *synchronisation Zs* du signal *Z* par l'horloge *h*, qui note à chaque instant d'horloge la valeur de *Z*. Cela permet d'obtenir des signaux *DS* et *FS* d'une durée constante égale à la période *h* (fig. 18.8 et 18.9).

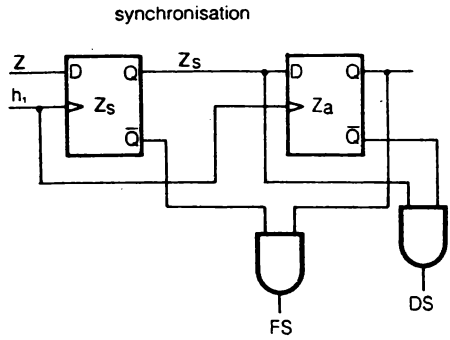


Fig. 18.8.

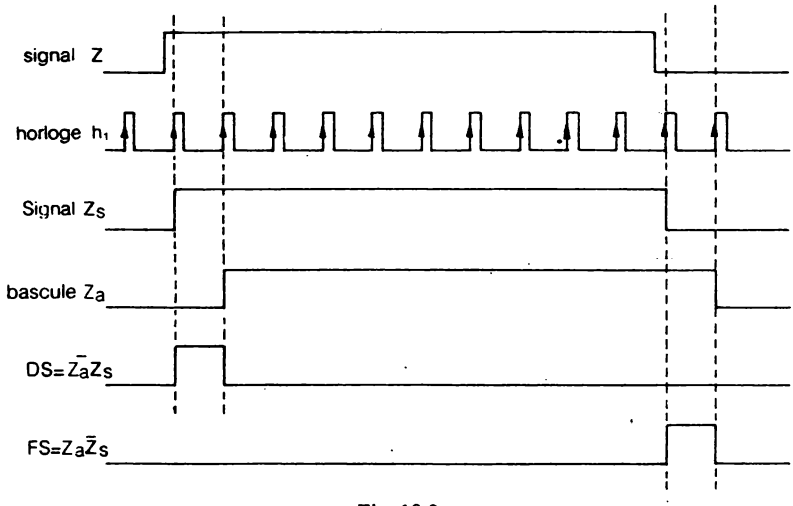


Fig. 18.9.

Le défaut que nous venons de mettre en évidence et corriger, dû à un manque de synchronisme, n'existe pas entre variables internes et horloges, mais est systématique entre variables externes et horloges. Aussi, toutes les variables externes devront être synchronisées par rapport au système séquentiel qui les exploitent. On utilise pour cela des bascules ou registres, commandés par une horloge du cycle interne au système.

4. SYSTÈME SIMPLE COMPOSÉ DE FONCTIONS QUELCONQUES: GÉNÉRATEUR DE CARACTÈRES

4.1. Spécification

On veut réaliser un générateur de caractères à quatre éléments binaire *DCBA*, en synchronisme avec une horloge *h* (fig. 18.10). La période du générateur est au maximum de 16 caractères.

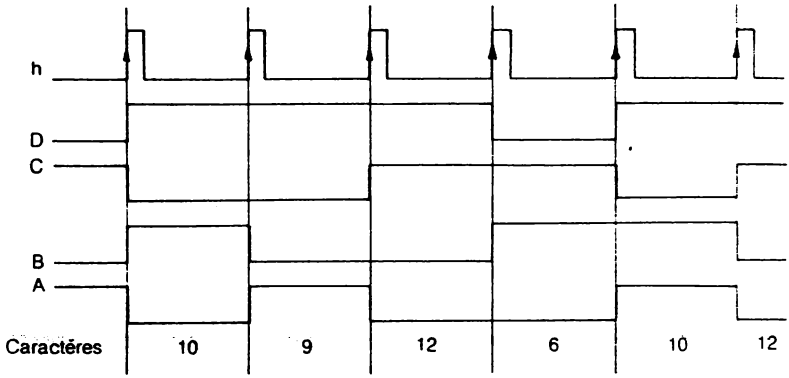


Fig. 18.10.

4.2. Principe de réalisation

Deux cas sont à considérer:

- Dans la séquence émise par le générateur on ne retrouve pas deux fois le même caractère. On peut alors calculer le prochain caractère à partir de l'état du caractère présent, selon un tableau de vérité. On obtient les fonctions D^+, C^+, B^+, A^+ à partir de D, C, B, A. On représente les quatre variables D, C, B, A par quatre bascules et leurs commandes par des portes.
- Dans la séquence émise par le générateur on retrouve plusieurs fois le même caractère. On ne peut plus appliquer le même principe: l'état du caractère suivant ne dépend plus du caractère présent mais de son rang dans la séquence. La réalisation est alors différente: avec un compteur programmé pour tourner sur N positions, on adresse une mémoire

PROM qui contient les N codes, ou encore on calcule les caractères à partir des positions du compteur.

4.3. Calcul d'une séquence sans répétition de caractère

Représentons les caractères par leur équivalent décimal. Soit à réaliser la séquence :

[15, 10, 12, 13, 7, 3, 2, 0, 11] [15, 10, ..., 11]

La figure 18.11 donne la table de vérité, la figure 18.12 les formules et la figure 18.13 la réalisation de principe avec des bascules D et des portes. Si ce schéma est réalisé avec des portes et des bascules *SSI*, la modification du cycle n'est pas aisée, car il faut recâbler d'autres portes. Par contre si l'on utilise des circuits programmables séquentiels (comprenant portes et bascules) il suffit de reprogrammer les fonctions. On obtient également un montage facilement modifiable en remplaçant le calcul des fonctions par une mémoire PROM : le caractère présent sert d'adresse à la PROM dans laquelle on a inscrit les successeurs des caractères. Un changement de séquence ne se traduit plus que par un changement de contenu de la PROM. En reprenant le tableau de vérité de la figure 18.11, on voit qu'à l'adresse 0 on écrit 11, à l'adresse 1 une valeur quelconque X, à l'adresse 2 la valeur 0, et ainsi de suite pour terminer à l'adresse 15 avec la valeur 10. On obtient le schéma de la figure 18.14.

	D	C	B	A	D^+	C^+	B^+	A^+	
0	0	0	0	0	1	0	1	1	11
1	0	0	0	1	X	X	X	X	X
2	0	0	1	0	0	0	0	0	0
3	0	0	1	1	0	0	1	0	2
4	0	1	0	0	X	X	X	X	X
5	0	1	0	1	X	X	X	X	X
6	0	1	1	0	X	X	X	X	X
7	0	1	1	1	0	0	1	1	3
8	1	0	0	0	X	X	X	X	X
9	1	0	0	1	X	X	X	X	X
10	1	0	1	0	1	1	0	0	12
11	1	0	1	1	1	1	1	1	15
12	1	1	0	0	1	1	0	1	13
13	1	1	0	1	0	1	1	1	7
14	1	1	1	0	X	X	X	X	X
15	1	1	1	1	1	0	1	0	10

Fig. 18.11.

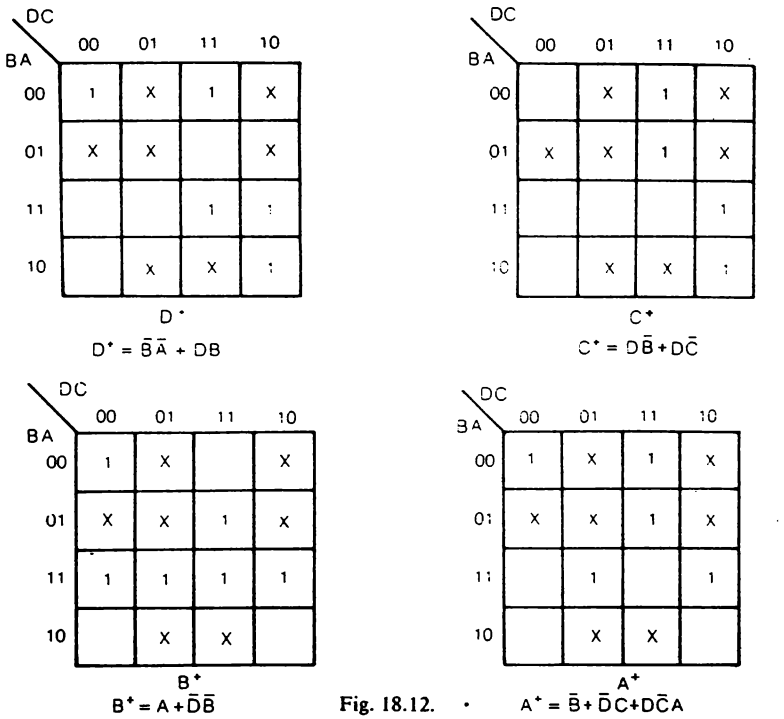


Fig. 18.12.

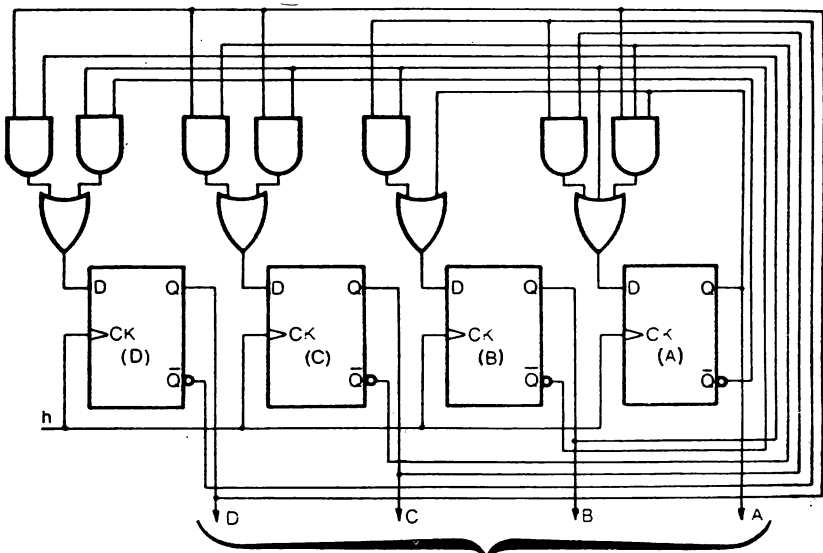


Fig. 18.13.

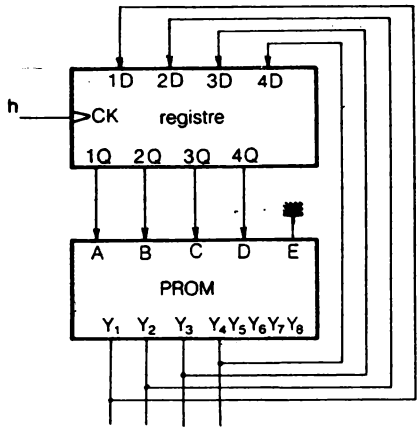


Fig. 18.14.

4.4. Cycle avec répétition

On se sert d'un compteur pour adresser les caractères successifs en mémoire PROM. La liste des caractères peut être quelconque (avec ou sans répétition). Quand on obtient le dernier caractère du cycle on remet à zéro le compteur d'adresse. On obtient le schéma de la figure 18.15.

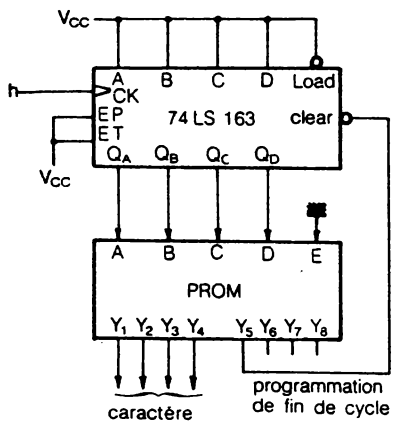


Fig. 18.15.

BIBLIOGRAPHIE

H. BOUCHER. — *Architecture de l'ordinateur*, (Cepadues Éditions).

J. FLORINE. — *Automatismes à séquences et commandes numériques*, (Dunod).

J. LAGASSE, M. COURVOISIER et J.-P. RICHARD. — *Logique combinatoire*, (Dunod Université).

J. LAGASSE, M. COURVOISIER et J.-P. RICHARD. — *Logique séquentielle*, (Dunod Université).

D. MANGE. — *Traité d'électricité*, Tome V, Analyse et synthèse des systèmes logiques, (Éditions Georgi).

J. CINAL. — *Techniques Booléennes et calculateurs arithmétiques*, (Dunod).

J.-M. BERNARD, J. HUGON, R. LE CORVEC. — *De la logique câblée aux microprocesseurs*, Tomes 1, 2, 3 et 4, (Éditions Eyrolles).

J.-M. BERNARD, H. BRETEUIL. — *Les circuits programmables*, (Éditions Test).

T.R. BLAKESLEE. — *Digital Design with standard MSI and LSI*, (Wiley).

AUMIAUX. — *Logique binaire et ordinateurs*, Tomes 1 et 2, (Masson).

Revue Electronics. — *Electronic Desing*, (Minis et micros).